CONTROL DATA

60461920

MSL 153/155 IOU
Test Descriptions

REFERENCE

Control Data
Computer Systems

MSL 153/155
IOU Test Descriptions

Reference

Publication Number 60461920

Manual History

Technical changes and additions are indicated by vertical change bars in the margins.

Revision	CIP Level	Date
A	Manual released.	May 1986
B	L678	April 1987
C	L688	September 1987
D	L700	April 1988
E	L710	August 1988
F	727	May 1989
G	CIP L757	September 1990
H	L765	January 1991
J	L780	August 1991
K	L780C	March 1992

Revision K of this manual adds support for the CYBER 970 and 972 computer systems. This edition obsoletes all previous editions.

Revision letters I, O, Q, S, X, and Z are not used.

Contents

Figures

About This Manual

This manual describes the CDC® CYBER 960, 962, 970, 972, 990, 990E, 992, 994, 995E, and 2000 Input/Output Unit (IOU) test. Procedures for using this test are provided in the MSL 153/155 IOU Test Procedures Reference Manual listed under Related Manuals.

This test description references level 1 through 3 diagrams (refer to AT478-A/B/C/D, AT481-A/B IOU and Associated Equipment Theory/Diagrams Hardware Maintenance Manual, and AT511-A/AT512-A and CYBER 2000 IOU and Associated Equipment Theory/Diagrams Hardware Maintenance Equipment, listed under Related Manuals). The terms and labels on these block diagrams have been used throughout the test documentation to provide the user with a means of quick and uniform reference.

NOTE

It is assumed and planned that the user of program descriptions will have an in-depth understanding of the processor logic; including familiarity with all levels of the Multi-Level Block Diagrams.

Conventions

The following conventions are used in this manual:

- Technical changes and additions are indicated by change bars and are correlated with the revision of the page on which they occur. Other changes, such as editorial and pagination, are not identified by change bars, but may be included as part of a revision.

- When the word "press" is used in an instruction, it tells you to press the named key or keys.

  For example, when you see the instruction:

  Press **Enter**

  you should press the key labeled **Enter** on your keyboard.

- When the word press precedes a string of key names that are separated by hyphens (for example **Ctrl-Alt-Del**), it means that you should press the keys listed simultaneously.

  For example, when you see the instruction:

  Press **Ctrl-Alt-Del**

  you should press the keys labeled **Ctrl**, **Alt**, and **Del** simultaneously.

- Boldface type is also used when identifying equipment panel nomenclature.

- User input and computer output are printed in a non-proportional typeface that simulates computer output.

  When the word "type" is used in an instruction, it tells you to type in the following word.

  For example, when you see the instruction:

  Type `TEST`

  you should type the letters `T E S T` in sequence.

Disclaimer

This product is intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features or undefined parameters.

Related Manuals

Control Data Publication	Publication Number
Maintenance Software Library (MSL 15X) Reference Manual	60456530
MSL 150 – Model Independent Test Descriptions	60469390
AT478-A/B/C/D, AT481-A/B IOU and Associated Equipment Theory/Diagrams Hardware Maintenance Manual	60463540
AT511-A/AT512-A and CYBER 2000 IOU and Associated Equipment Theory/Diagrams Hardware Maintenance Equipment	60000235
Standardized Maintenance Approach Quick Reference (SMAQR)	60462110
CYBER 960 Series and 970 Series System Troubleshooting Guide	60000122
CYBER 2000 System Maintenance Guide	60000334
CYBER 960 Series and 970 Series Maintenance Software Reference	60000292
CYBER 180 Model 990 Maintenance and Parts	60462150
CYBER 180 Model 990 Central Memory Maintenance and Parts	60458260
CIP Reference Manual, CYBER 180 Model 810, 830, 815, 825; CYBER 810A, 830A Computer Systems	60000417
CIP Reference Manual, CYBER 180 Model 835, 845, 855; CYBER 840, 850, 860 Computer Systems with IOU AB115A	60000418
CIP Reference Manual, CYBER 180 Model 845, 855; CYBER 840, 850, 860 With IOU AT478A/AT481A; CYBER 840A, 850A, 860A, 870A, 990, 990E, 995E Computer Systems	60000419
CIP Reference Manual, CYBER 960, 970, 994 Computer Systems	60000420
CIP Reference Manual, CYBER 962, 972, 992 Computer Systems	60000421
CIP Reference Manual, CYBER 170 Model 865, 875; Non-Model 8XX/9XX Series Computer Systems	60000422
CYBER 2000 System Console User Interface/CIP Reference Manual	60000539
CYBER 2000 CP/CM Test Procedures\Descriptions Reference Manual	60000542
MSL 153/155 IOU Test Procedures Reference Manual	60461110

Control Data Publication	Publication Number
ACT5 – MSL 155 Test Description	60461120
BPT5 – MSL 155 Test Description	60461130
CMT5 – MSL 155 Test Description	60461140
EPT5 – MSL 155 Test Description	60461150
FPT5 – MSL 155 Test Description	60461160
IDT5 – MSL 155 Test Description	60461170
IFT5 – MSL 155 Test Description	60461180
IGT5 – MSL 155 Test Description	60461190
LST5 – MSL 155 Test Description	60461200
MAT5 - MSL 155 Test Description	60461210
MIT5 – MSL 155 Test Description	60461220
OCT5 – MSL 155 Test Description	60461230
PAT5 – MSL 155 Test Description	60461240
SCT5 – MSL 155 Test Description	60461250
VAT5 – MSL 155 Test Description	60461260
MST5 – MSL 155 Test Description	60461270
ANT3/P – CYBER 960 Series and 970 Series MSL Test Description	60000278
BPT3/P – CYBER 960 Series and 970 Series MSL Test Description	60000279
CMT3/P – CYBER 960 Series and 970 Series MSL Test Description	60000280
CST3/P – CYBER 960 Series and 970 Series MSL Test Description	60000281
CTT3/P – CYBER 960 Series and 970 Series MSL Test Description	60000282
ICT3/P – CYBER 960 Series and 970 Series MSL Test Description	60000283
IFT3/P – CYBER 960 Series and 970 Series MSL Test Description	60000284
LMT3/P – CYBER 960 Series and 970 Series MSL Test Description	60000285
MAT3/P – CYBER 960 Series and 970 Series MSL Test Description	60000286
MCT3/P – CYBER 960 Series and 970 Series MSL Test Description	60000287
MDT3/P – CYBER 960 Series and 970 Series MSL Test Description	60000288
OIT3/P – CYBER 960 Series and 970 Series MSL Test Description	60000289
PDT3/P – CYBER 960 Series and 970 Series MSL Test Description	60000290
SMT3/P – CYBER 960 Series and 970 Series MSL Test Description	60000291

Submitting Comments

Control Data welcomes your comments about this manual. Your comments may include your opinion of the usefulness of this manual, your suggestions for specific improvements, and the reporting of any errors you have found.

You can submit your comments on the comment sheet on the last page of this manual. If the manual has no comment sheet, mail your comments to:

Control Data
Technical Publications ARH219
4201 Lexington Avenue N.
St. Paul, MN 55126-6198

Please indicate whether you would like a written response.

If you have access to SOLVER, the Control Data facility for reporting problems, you can use it to submit comments about the manual. When entering your comments, use NV0 (zero) as the product identifier. Include the name and publication number of the manual.

Ordering Manuals

To order a Control Data manual, send an order form to:

Control Data
Literature and Distribution Services ARHLDS
4201 Lexington Avenue N.
St. Paul, MN 55126-6198

To obtain an order form or to get more information about ordering Control Data manuals, write to the above address or call (612) 482-3800 or 482-3801, or FAX your inquiry to (612) 482-3813. If you are a Control Data employee, use the Controlnet number 235-3800, 235-3801, or 235-3813.

AT478-A/B/C, AT481-A/B IOU and Associated Equipment 1

AT478-A/B/C, AT481-A/B IOU and Associated Equipment 1

LDS – Long Deadstart Sequence

The Long Deadstart Sequence (LDS) is a quicklook program that resides in read-only memory (ROM) attached to upper core of logical PP00 memory (6000 through 7777_8). It is possible to select any physical peripheral processor (PP) as logical PP00 by entering the RB x and RP xx commands from the keyboard. Initiating long deadstart sequence sets the following conditions:

- A master clear is issued to all PPs and all error status bits cleared.
- PP00 is forced to read the test program stored in ROM.
- PP00 address is forced to 6000_8.
- A PP register display is brought up on the right-hand side of the screen.

While LDS is executing, certain error status bits are monitored to determine if the hardware failure has been detected. If a hardware failure is detected, a message is displayed below the PP register. If LDS does not detect a hardware failure, the Extended Deadstart Sequence (EDS4) is loaded and executed.

Each error stop is indicated by a unique P register address. The customer engineer (CE) determines what error condition was detected by examining the P register display and referring to the error code directory or to Structured Analysis Method (SAM) lists. Refer to the appropriate Standardized Maintenance Approach Quick Reference (SMAQR) manual.

When execution of LDS is finished, the Short Deadstart Sequence is initiated using the program currently displayed on the screen.

LDS TESTS

LDS performs the following sequence of tests:

1. Arithmetic Unit Test
2. PP00 Memory Test (up to location 5777_8)
3. One Word Channel Test
4. Block I/O Transfer Test
5. Block I/O Transfer with Conversion Test
6. PP01 Through PP04 Memory Test
7. Multi-PP Arithmetic Unit Test
8. LDS Bit Test

The following paragraphs briefly describe these tests.

Arithmetic Unit Test

The tests performed by the Arithmetic Unit Test are described in the following table.

Test	Description
Unconditional Jump Test	Checks execution of the 03 instruction and the lower 6 bits of the d portion of the instruction. Error stops are: - on 0300 or 0377 instruction for single Q bit failure. - indicated by hardware timeout counter error for a multiple Q-bit failure or P register failure.
Conditional Jump Test	Checks instructions 04 through 07 with A equal to 0, A equal to 1, and A equal to minus 0. Error stops are on 03 through 07 instructions.
1x Instructions Test	Checks instructions 10 through 17 using lower 6 bits of A register. Error stops are on branch instructions.
2x Instructions Test	Checks instructions 20 through 25, arithmetic operations in A with all 18 bits, and shift network. Error stops are on branch instructions.
3x Instructions Test	Checks instructions 30 through 37 using lower addresses of PP00 memory for store instructions. Error stops are on branch instructions.
4x Instructions Test	Checks instructions 40 through 47 and all 16 bits of Q register. Lower addresses of PP00 memory is used to check restore capability. Error stops are on branch instructions.
5x Instructions Test	Checks instructions 50 through 57 with and without Q flag set. Error stops are on branch instructions.
13x Instructions Test	Checks execution of instructions 130 through 137 with 16-bit operands. Error stops on branch instructions.
14x Instructions Test	Checks execution of instructions 140 through 147 with 16-bit operands. Error stops are on branch instructions.
15x Instructions Test	Checks execution of instructions 150 through 157 with 16-bit operands, with and without Q flag set. Error stops are on branch instructions.
RJM and LJM Instructions	Check execution of instructions. Test 01 and 02 using previously written addresses in lower core of PP00 memory. Error stops are on branch instructions.

PP00 Memory Test

This test executes from low addresses of PP00 memory. The test is transferred from ROMs to PP memory and checked for correct transfer. Error stop is on a branch instruction.

The test then checks PP00 memory with 10 fixed data patterns and addresses. Error stop is on branch instruction. The contents of the A register represent the failing bits. The failing pattern and failing address are stored in low addresses of PP00 memory. The following table indicates the bit locations in PP00 memory.

PPM00 Location	Contents
34_8	Failing bits
35_8	Failing patterns
36_8	Failing addresses

One Word Channel Test

This test checks channels 00 through 04 and 17 with one word input/output (I/O) using all possible bit combinations and channel status during I/O. These channels are tested because they are present with any IOU configuration and are essential for successful running of Common Maintenance Software Executive (CMSE). Error stops are on branch instructions and are unique for each channel status and data error.

Block I/O Transfer Test

This test checks inter-PP communication over the channels using a slightly modified PP memory test. A timeout counter protects PP00 from a hung condition. Error stops are on branch instructions that are unique for each channel/PP.

Block I/O Transfer with Conversion Test

This test checks inter PP communication with 12- to 16-bit and 16- to 12-bit conversion over channels 01 through 04. Error stops are on branch instructions. A timeout feature protects PP00 from a hung condition.

PP01 Through PP04 Memory Test

This test checks PP01 through 04 memories with ten fixed patterns and addresses. Error stops are in PP00 on branch instructions. The CE is able to find failing bits, failing patterns, and failing addresses that are stored in the following locations in PP00 memory.

PPM00 Location	Contents
34_8	Failing bits
35_8	Failing patterns
36_8	Failing addresses

1

Multi-PP Arithmetic Unit Test

This test is the same as the Arithmetic Unit Test described previously, but runs in PP01 through PP04 simultaneously. If an error is detected, the PP in which the test is running stops and P register stop lights are the same as $ for the Arithmetic Unit Test. PP00 is in a loop waiting for a message from PPx indicating the test successfully passed. The looping address is unique to each PP.

LDS Bit Test

This test checks the LDS bit in the Maintenance register. If the bit is clear, PP00 stops and the P register indicates an error in the Maintenance register. EDS4 loading is initiated if the LDS bit is set and EDS4 is selected.

EDS4 - Extended Deadstart Sequence

EDS is similar in structure to LDS. It checks the hardware needed by CMSE that was not tested by LDS. When LDS completes, the common test and initialization (CTI) package loads EDS4 into PP00 from the CTI device if bit 2^0 of word 12 is set in deadstart panel program.

Each error stop is indicated by a unique P register address. The CE determines which error condition was detected by examining the P register and referring to the SAM lists.

EDS4 Tests

EDS4 performs the following sequence of tests:

1. Real Time Clock Test
2. Initial Maintenance Register Test
3. Channel 10_8 and 15_8 Test
4. Channel Flags Test

All tests are executed from PP00 and all error stops are on branch instructions. The following paragraphs briefly describe these tests.

Real Time Clock Test

This test checks operation of channel 14_8, the real time clock channel, with all possible clock readings (0 through 7777_8) to assure the clock is working properly.

Initial Maintenance Register Test

This test checks access to the IOU Maintenance registers and the hardware features that are used in other tests under CMSE to assure that those features are operational.

Channel 10 and 15 Test

This test is performed only if these channels are installed, which is determined by reading the Options Installed register.

The test consists of one-word I/O test over channels 10_8 and/or 15_8 using 100_8 different patterns, and a channel status test. Error stops are on branch instructions. PP10 is forced to the idle condition before the one-word channel test is started.

Channel Flags Test

This test checks channel flags for the channels that have been installed in the IOU. CMSE communication structure requires successful completion of this test.

QLT4 - Quick Look Test

The Quick Look Test tests one-word I/O over each channel. The test uses hardware features such as force idle PP and force deadstart PP.

At the end of the test, all tested channels are set active, empty, and all associated PPs, if selected, are in deadstart condition.

Section Descriptions

The Quick Look Test section numbers, converted to octal, represent the channel under test (that is, section 8 represents channel 10_8). For a channel to be tested, both the section and the channel must be selected by bits in the section select and channel flag parameter words. Refer to parameters and control words in paragraph 3 in this part of the manual.

This section has the following four subsections.

Subsection	Tag	Description
00	SUBSEC00	Initializes the associated PP (force idle PP).
01	SUBSEC01	Tests one word I/O over tested channel.
02	SUBSEC02	Tests channel status test with escape bit (Q bit 2^5) set.
03	SUBSEC03	Forces PP in deadstart condition.

At the end of the section, channel status is active, empty, and the associated PP is in deadstart condition.

Subsection 00 (SUBSEC00)

Subsection 00 has two conditions that are executed only if the associated PP for the channel has been selected.

Condition	Description
00	Forces PP idle through the Maintenance register, reads K register of the associated PP, and checks for 1077_8 indicating that the PP is idle.
01	Tests channel flag operation.

Subsection 01 (SUBSEC01)

Subsection 01 tests one word I/O over each channel selected to be tested and checks the status of the channel during the I/O transfer. This subsection has 1000_8 conditions, that use different patterns and complements of that pattern. There are 152_8 fixed patterns and 626_8 randomly generated patterns.

Each condition uses one pattern and the complement of the pattern. During this I/O, the test checks for correct channel status and channel error flag clear status.

Subsection 02 (SUBSEC02)

Subsection 02 has the following nine conditions.

Condition	Description
00	Disconnects already inactive channel with escape bit set.
01	Outputs one word on inactive channel with escape bit set.
02	Inputs one word from inactive channel with escape bit set.
03	Blocks input over inactive channel expecting that the first word address of the block input is cleared.
04	Blocks output over inactive channel expecting that the contents of A are not changed.
05	Activates already active channel with escape bit set.
06	Functions FAN on active channel with escape bit set.
07	Functions FNC on active channel with escape bit set.
08	Forces channel active and full and checks for empty when channel is deactivated.

NOTE

It is possible for subsection 02 to cause the PP from which the test is being executed to hang if the escape feature (bit 2^5 of I/O instructions) does not work. There is no way to prevent this hang if the feature fails.

Subsection 03 (SUBSEC03)

This subsection has two conditions. Condition 01 is executed only if the channel has an associated PP and PP is selected.

Condition	Description
00	Tests channel for active and empty status.
01	Forces PP to deadstart condition. Check P register of the associated PP for 7777_8 value. Check K register of the associated PP for 71_8 value. Check Q register of the associated PP for channel number. Check A register of the associated PP for 10000_8 value.

1

PMT4 – PP Memory Test 1

PP Memory Test 1 (PMT4) checks the operation of the PP memories in all barrels not used by the IOU Control Program (IOUCP). The test does not rely on code executing in the NIO PP memory being tested. A CIO PP must be able to execute a small program to switch the I/O channel from channel 17_8 to channel 15_8. PP memories are tested sequentially using force deadstart, idle, and force dump. Since the barrel being used by IOUCP was checked by the LDS/EDS4 tests, PMT4 assumes that it is operational.

Section Descriptions

The PP Memory Test 1 section numbers, in decimal, represent the PP memory being tested. For a memory to be tested, both the section and the PP must be selected by bits in the section select and PP flag parameter word. Refer to parameters and control words in paragraph 3 in this part of the manual.

All data transfers for NIO PPs are done over one NIO channel, with the capability to switch channels if a channel error is detected. All data transfers for CIO PPs are done over channel 17_8, except for the deadstart and loading of the program to switch the I/O, when channel 15_8 is used.

This section has 32 subsections numbered 00 through 31, as described in the following table.

Subsection	Tag	Description
00	B00.00	Forces deadstart PP with check.
01	B01.00	Checks P and A registers.
02 to 30	B02.00	Tests memory with patterns.
31	B03.00	Tests memory with address patterns.

At the beginning, all PPs are forced into a hung condition by a force idle feature. The PPs are left idle at the end of each section, but are forced to a deadstart condition at the end of the test.

Subsection 00 (B00.00)

This subsection has the following five conditions.

Condition	Description
00	Forces PP to deadstart condition.
01	Checks P register of tested NIO PP for the value 177777_8 or checks P register of tested CIO PP for the value 000001_8.
02	Checks Q register of tested NIO PP for deadstart channel value or checks Q register of tested CIO PP for the value 000016_8.
03	Checks K register of tested NIO PP for the value 007100_8 or checks K register of tested CIO PP for the value 107700_8.
04	Checks A register of tested NIO PP for the value 020000_8 or checks A register of tested CIO PP for the value 020000_8.

Subsection 01 (B01.00)

This subsection has the following condition.

Condition	Description
00	Repeats one word output to fill tested memory. Checks the P and A registers of the NIO PP being tested for correct count after each word of output. The P and A registers in the CIO PP being tested are not checked.

Subsections 02 through 15 (B02.00)

These subsections output standard test patterns to complement every second word, then dump the test PP over the used channel to the monitor PP to check the data.

These subsections have the following two conditions.

Condition	Description
00	Outputs data blocks of 100_8 words to fill up the test PP memory.
01	Forces a dump of the test PP and input one word at a time checking data.

Subsections 16 through 30 (B02.00)

These subsections use the same code as subsections 01 through 15, but use random number patterns.

Subsection 31 (B03.00)

This subsection outputs the address of the corresponding memory location to fill the test PP memory. Then dump the test PP over the used channel to the monitor PP to check the data.

This subsection has the following two conditions.

Condition	Description
00	Repeats one word output to fill the memory of the test PP.
01	Forces a dump of the test PP and input one word at a time checking data.

1

EXT4 - Execution Unit Test

The Execution Unit Test checks instruction execution and the arithmetic units in PPs other than the one in which the IOUCP and CMSE are running. CMSE loads EXT4 into the same PP with IOUCP, inserts the communication channel in the communication routine, and distributes the updated copies to the PPs to be tested. The test is started by deactivating the communication channel for the assigned PP.

Section Descriptions

EXT4 is divided into the following three sections.

Section	Tag	Description
00	UJNTST	Tests arithmetic unit and execution of instructions 00 through 57_8 (12-bit operand instructions).
01	TEST13X	Tests arithmetic unit and execution of instructions 1030 through 1057_8 (16-bit operand instructions).
02	STSTEST	Tests execution of the channel instructions.

Section 00 (UJNTST)

This section tests 12-bit operand instructions 00 through 57_8 and is divided into the following six subsections.

Subsection	Tag	Description
00	UJNTST	Tests execution of 00 through 07 instructions.
01	TEST1X0	Tests execution of 10 through 17 instructions and A register bits 2^0 through 2^5.
02	TEST2X0	Tests execution of 20 through 23 instructions and A register bits 2^0 through 2^{17}.
03	TEST3X	Tests execution of 30 through 37 instructions and restore ability.
04	TEST4X	Tests execution of 40 through 47 instructions and A register bits 2^0 through 2^{11}.
05	TEST5X	Tests execution of 50 through 57 instructions and Q adder.

Section 01 (TEST13X)

This section tests 16-bit operand instructions and is divided into the following four subsections.

Subsection	Tag	Description
00	TEST13X	Tests execution of 1030 through 1037 instructions and restore ability on 16-bit boundary.
01	TEST14X	Tests execution of 1040 through 1047 instructions and restore ability using indirect addressing.
02	TEST15X	Tests execution of 1050 through 1057 instructions using index addressing.
03	JMPTEST	Tests execution of 01 and 02 instructions.

Section 02 (STSTEST)

This section tests I/O instructions (channel instructions) and has the following four subsections.

Subsection	Tag	Description
00	STSTEST	Tests 64 through 67 instructions.
01	IANTEST	Tests 70, 72, 74, and 75 instructions.
02	IOTEST	Tests 1064 through 1067 instructions.
03	ESCTEST	Tests channel instructions with escape bit (Q bit 2^5) set.

1

PMU4 – PP Memory Test 2

PP Memory Test 2 (PMU4) performs a more rigorous check of the PP memories than PMT4. The advantage over PMT4 is that PMU4 sends copies of the test code to the PP memories to be checked. This provides more variety in the way memory is referenced and allows more strenuous code to be used.

PMU4 allows the parameters to be changed for each copy to be loaded to a test PP, provided parameter change capability is chosen initially.

Section Descriptions

The section, subsection, and condition structure of PMU4 is defined in reference to where the code executes in a particular test PP. The code in the test PP is executable from either the lower or half of memory, so that the entire memory can be tested.

Selection of sections 00 and 01 tests the entire memory.

Section	Description
00	Executes in the lower half of memory and tests all memory except test core and direct cells used by the lower half copy.
01	Executes in the upper half of memory and tests direct cells and core used by the lower half copy.

Each section includes all of the following 48 subsections except as noted.

Subsection	Tag	Description
00	SS00	Checks addressing to find a solid addressing failure using a quicklook check. This is not affected by the memory bounds of a section.
01 through 36	SS01 through 36	Checks available test memory using standard and random patterns.
37 through 39	SS37 through 39	Tests available test memory with a random word in each location.
40 through 46	SS40 through 46	Loads random addresses rapidly in a prearranged sequence to check addressing.
47	SS47	Checks if faults are induced when address lines are selectively toggled. This is not affected by memory bounds of a section.

Subsection 00

This subsection uses the following two conditions to check for solid address failure.

Condition	Description
00	Initializes location 0 with all zeros, and stores ones in all remaining locations with one-bit-set address. A nonzero word in location 0 indicates an address line shorted.
01	Initializes location 7777_8 with all ones, and stores zeros in all remaining locations with one-bit-clear address. A zero-bit in location 7777_8 indicates an open address line.

Subsections 01 through 06

These subsections use a standard fixed pattern. The testable memory is checked in groups of five words with the pattern, then its complement, being loaded into alternate locations. The words are quickly read back and checked in sequential order.

Subsections 07 through 14

These subsections are the same as subsections 01 through 06, except the standard patterns are circularly shifted. These sliding patterns use 16 conditions, one for each circular shift.

Subsections 15 through 21

These subsections are exactly the same as subsections 01 through 06. The placement allows staggering of fixed and sliding pattern checks.

Subsections 22 through 36

These subsections are the same as subsections 01 through 06, except the patterns are randomly chosen before the tests are copied to the test memories.

Subsections 37 through 46

These subsections have one condition. Each subsection uses a real-time-clock input to generate a series of 1000_8 addresses. For each address the previous address is used as data. This daisy chain is then rapidly retraced using load instructions. If no error is generated by rapid changes of the address/data lines, the 1000th load is the reference point.

Repeating the condition uses the same seed for the random number generation; repeating the subsection uses a new seed for each cycle.

The subsections are duplicated to allow multiple cycles without selecting a repeat subsection.

Subsection 47

This subsection checks if the address lines are affected by crosstalk. An instruction sequence is executed that causes the address lines to switch from all zeros to all ones, except for one bit to be checked for staying zero. This is repeated for all 13 address bits then another sequence is used for checking all ones toggling to all (but one) zeros.

This subsection uses the following two conditions.

Condition	Description
00	Toggles all address lines, except the test bit, from zeros to ones by executing a store with indirect addressing through location 0 that contains a one-bit-clear address.
01	Toggles all address lines, except the test bit, from ones to zeros. This is accomplished by executing an indexed long jump instruction, from location 7776_8 to location 0 plus a one-bit-set address.

CHD4 - Channel Test

The Channel Test (CHD4) checks inter-PP data transfer over each tested channel.

The test uses a pair of PPs for inter-PP transfer. One PP is called the PP transmitter and the other, the PP receiver. The PP transmitter communicates through the Double PP Driver (DPPD) with the IOUCP over the communicating channel. The PP receiver communicates with the PP transmitter over the testing channel.

The PP transmitter and the PP receiver are randomly selected at the beginning of the test and whenever the test is repeated.

The execution of the complete test with one channel is considered a section. The number of selected sections specifies the number of the channels to be tested for one PP pair (default is 32 sections selected).

When all selected sections are done, the end of the test is reported and, if repeat test is selected, new pairs are randomly generated.

The maximum number of words (size of the I/O block) to be transferred is 1000_8.

Section Descriptions

In this test, a section is the execution of the complete program using one pair of PPs that are testing one channel. The PP receiver returns received data over the channel being tested to the PP transmitter. The PP-transmitter section is divided into the following five subsections.

Subsection	Tag	Description
00	SUBSEC00	Tests one word I/O over tested channel using OAN instruction.
01	SUBSEC01	Tests one word I/O over tested channel using OAM instruction.
02	SUBSEC02	Tests the channel flag operation (PP-receiver number is the channel number for channel flag test).
03	SUBSEC03	Tests block I/O over tested channel using a OAM instruction.
04	SUBSEC04	Tests block I/O over tested channel using a OAPM instruction (conversion).

Subsection 00 (SUBSEC00)

Subsection 00 tests the data and channel status during one word I/O using a OAN (single word output) instruction over the channel being tested. The data used for I/O are word counts. Every condition is represented by one word count.

Subsection 01 (SUBSEC01)

Subsection 01 tests the data and channel status during one word I/O using a OAM (output block) instruction over the channel being tested. The data used for I/O are data patterns used in subsections 03 and 04. Every condition represents one pattern.

Subsection 02 (SUBSEC02)

Subsection 02 cannot be deselected. If subsections 03 or 04 are selected, then subsection 02 is selected. It tests the operation of the channel flag (channel flag number is equal to the receiving PP number). This channel flag is used in the subsections 03 and 04 to control the receiving PP.

Subsection 03 (SUBSEC03)

Subsection 03 tests the data and channel status during block I/O using a output block (OAM) instruction over the channel being tested. The size of the block is represented by the word count. Every condition is one block size (one word count).

Subsection 04 (SUBSEC04)

Subsection 04 tests the data and channel status during block I/O using output block (OAPM) instruction with conversion over the channel being tested. The size of the block is represented by the word count. Every condition is one block size (one word count).

CMA4 - Central Memory Access Test

The Central Memory Access Test (CMA4) checks the data path from PP memory to central memory (CM) and from CM to PP memory. It also checks the Relocation (R) register which, in conjunction with the A register, makes the CM address.

Section Descriptions

CMA4 is divided into ten sections.

Section	Tag	Description
00	SEC00	Tests R register.
01	SEC01	Tests (R+A), CM address.
02	CWDTST	Tests read/write CM with 60-bit single CM word.
03	ADRTST	Tests read/write addresses in CM with 60-bit single CM word.
04	CWMTST	Tests read/write CM with 64-bit single CM word.
05	CLDTST	Tests read/write block in CM with 60-bit CM word.
06	CLMTST	Tests read/write block in CM with 64-bit CM word.
07	RSLTST	Tests read and set lock.
08	RCLTST	Tests read and clear lock.
09	MIXTST	Tests read/write in mixed mode (60- and 64-bit CM word, single CM word and block).

Section 00 (SEC00)

This section has the following three subsections.

Subsection	Tag	Description
00	RRG00	Tests the lower 12 bits of R register, with the upper 10 bits of R set to zero.
01	RRG01	Tests the upper 10 bits of R register, with the lower 12 bits of R set to zero.
02	RRG02	Tests full R register.

Every subsection has 1000_8 conditions and every condition is a different pattern.

Section 01 (SEC01)

This section has the following subsection.

Subsection	Tag	Description
00	RAATST	Tests full A+R register.

This subsection has 1000_8 conditions and every condition is a different pattern.

Sections 02 through 09

Sections 02 through 09 have 16 subsections each. Subsections 00 through 07 use different starting addresses in CM and do not use the R register. Subsections 08 through 15 use randomly selected R-register values.

Every subsection has 1000_8 conditions and every condition is a different pattern with a different starting address in CM.

For the block read/write, every condition is a different CM word count.

MRA4 – Maintenance Register Access Test

The Maintenance Register Access Test (MRA4) checks Maintenance register (MR) access from all PPs. It also checks the Maintenance Channel (MCH) hardware interlock and MCH priority circuits.

Section Descriptions

MRA4 has the following 26 sections.

Section	Tag	Description
00	SEC00	Tests the MCH interlock and PP priority.
01 through 25		Tests the MCH access from all PPs.

Section 00 (SEC00)

This section has the following two subsections.

Subsection	Tag	Description
00	S00B00	Tests that a PP from the lower barrel has priority over other PPs.
01	S00B01	Tests that only one PP can access the MCH at a time, using MCH interlock.

Sections 01 through 25

Each section tests a MR access from the corresponding PP. Each section has the following five subsections.

Subsection	Tag	Description
00	FUNRES	Tests the response on function. Each condition is one function code.
01	ECHO	Tests the data to and from MR using an Echo function. Each condition is one pattern.
02	WRITE	Tests the Write function. The testing PP writes MR and the monitor PP reads and compares values.
03	READ	Tests the Read function. Each condition is one register in MR.
04	SSUMF	Tests the Status Summary function by reading data using the Status Summary function and normal read operation.

MRT4 – Maintenance Register Test

The Maintenance Register Test (MRT4) checks all parity networks in the IOU using invert parity feature. All the force conditions that induce errors to other parts of the hardware are also checked.

The 24-digit expected and received messages are the expected and received contents of Fault Status 1 and Fault Status 2 registers.

Section Descriptions

Each PPU is a section in test MRT4.

Subsection Descriptions

MRT4 is divided into the following 48 subsections.

Subsection	Tag	Description
00	SUB00	Tests for no reported errors.
01	SUB01	Tests invert parity on channel data to PPU.
02	SUB02	Tests invert parity from PPU to channel.
03	SUB03	Tests invert PPM to r parity.
04	SUB04	Tests invert parity at PPM data checker.
05	SUB05	Tests invert microcode parity.
06	SUB06	Tests invert PPM parity at parity generator.
07	SUB07	Tests invert parity on CM function code.
08	SUB08	Tests invert parity on Y register at parity generator.
09	SUB09	Tests invert parity on A register at parity generator.
10	SUB10	Tests invert parity on shift ROM at parity checker.
11	SUB11	Tests invert parity on Q register at parity generator.
12	SUB12	Tests invert parity on P register at parity generator.
13	SUB13	Tests invert parity on G register at parity generator.
14	SUB14	Tests invert R register to Y parity at checker.
15	SUB15	Tests invert PPM address parity at checker.
16	SUB16	Tests invert tag-out parity at generator.
17	SUB17	Tests invert data-in parity at generator.
18	SUB18	Tests invert CM address parity.

Subsection	Tag	Description
19	SUB19	Tests disable single-error-correction/double-error-detection (SECDED) code to PP memory.
20	SUB20	Not used.
21	SUB21	Not used.
22	SUB22	Not used.
23	SUB23	Not used.
24	SUB24	Tests force CM request/resync error.
25	SUB25	Tests force mark line parity bit to zero.
26	SUB26	Tests force tag-out/in parity error.
27	SUB27	Tests force response code parity error.
28	SUB28	Tests invert function-out parity bit.
29	SUB29	Tests force address-out parity bit low.
30	SUB30	Tests force data-in parity bit low.
31	SUB31	Tests force data-out parity bit low.
32	SUB32	Tests invert channel parity from the two-port mux or the maintenance channel.
33	SUB33	Tests invert Maintenance register write parity.
34	SUB34	Tests invert nanocode parity.
35	SUB35	Tests invert read parity bit.
36	SUB36	Tests invert channel 15 data-bus parity at checker.
37	SUB37	Tests invert R/I read-data parity.
38	SUB38	Tests invert R/I write-data parity.
39	SUB39	Tests force lost clock error.
40	SUB40	Tests force CM request in the ADU.
41	SUB41	Tests force tag-in parity error in the ADU.
42	SUB42	Tests force response code parity error in the ADU.
43	SUB43	Tests block d5 full in the ADU.
44	SUB44	Tests force CMC busy at CMI.
45	SUB45	Tests invert O. S. bounds address parity at checker.
46	SUB46	Tests set inhibit PPU to CM request.
47	SUB47	Tests clear inhibit PPU to CM request.

1

NOTE

Subsections 46 and 47 must be run together because subsection 46 sets up the condition that subsection 47 clears. Therefore, you can not select one section without selecting the other.

Subsection 00 (SUB00)

This subsection tests that no errors are reported when a code of (00) is forced in the Test Mode register.

Subsection 01 (SUB01)

This subsection tests for invert parity on a channel to PPU data transfer.

Subsection 02 (SUB02)

This subsection tests for invert parity on a PPU to channel data transfer.

Subsection 03 (SUB03)

This subsection is executed to cause parity error for PPU to R register.

Subsection 04 (SUB04)

This subsection is executed to cause parity error at the PPU memory data checker.

Subsection 05 (SUB05)

This subsection causes parity errors in microcode and the G register.

Subsection 06 (SUB06)

This subsection tests the parity network for data sent to the PP memory.

Subsection 07 (SUB07)

This subsection tests the parity error on CM function.

Subsection 08 (SUB08)

This subsection tests the parity network of the Y register.

Subsection 09 (SUB09)

This subsection tests the A register parity error network.

Subsection 10 (SUB10)

This subsection shifts the data in A to cause shift control ROM parity error.

Subsection 11 (SUB11)

This subsection shifts the data for checking parity in the Q register.

Subsection 12 (SUB12)

This subsection cause parity errors in the P register.

Subsection 13 (SUB13)

This subsection causes a parity error in microcode and the G register.

Subsection 14 (SUB14)

This subsection is executed to cause parity error for R-to-Y transfer.

Subsection 15 (SUB15)

This subsection causes parity errors in the PPU memory address at the parity checker.

Subsection 16 (SUB16)

This subsection tests the data to CM with inverted tag-out parity.

Subsection 17 (SUB17)

This subsection tests data-in parity on CM transfers.

Subsection 18 (SUB18)

This subsection tests the parity error reporting when CM address parity bit is inverted.

Subsection 19 (SUB19)

This subsection tests the disable SECDED code generator to PP memory.

Subsection 20 (SUB20)

This subsection code is not used.

Subsection 21 (SUB21)

This subsection code is not used.

Subsection 22 (SUB22)

This subsection code is not used.

Subsection 23 (SUB23)

This subsection code is not used.

Subsection 24 (SUB24)

This subsection tests for the CM request/resync error.

Subsection 25 (SUB25)

This subsection tests the force mark line parity bit low.

Subsection 26 (SUB26)

This subsection tests the force tag-out/in parity error.

Subsection 27 (SUB27)

This subsection tests the force response code parity error.

Subsection 28 (SUB28)

This subsection tests the invert function-out parity bit.

Subsection 29 (SUB29)

This subsection tests the force address-out parity bits low.

Subsection 30 (SUB30)

This subsection tests the parity error reporting when data-in parity is forced low on all groups.

Subsection 31 (SUB31)

This subsection tests the force data-out parity bit low.

Subsection 32 (SUB32)

This subsection tests the invert channel parity on channel 17_8, or when the parity is inverted on channel 15_8 data bus.

Subsection 33 (SUB33)

This subsection tests the invert Maintenance register write parity on the data being written into the register.

Subsection 34 (SUB34)

This subsection tests the invert nanocode parity.

Subsection 35 (SUB35)

This subsection tests the invert read parity bit.

Subsection 36 (SUB36)

This subsection tests parity error reporting when the parity is inverted on channel 15 data bus.

Subsection 37 (SUB37)

This is the test for radial interface read.

Subsection 38 (SUB38)

This is the test for radial interface write.

Subsection 39 (SUB39)

This subsection tests the force lost clock error.

Subsection 40 (SUB40)

This subsection tests the force CM request at the ADU.

Subsection 41 (SUB41)

This subsection tests the force tag-in parity error at the ADU.

Subsection 42 (SUB42)

This subsection tests the force response code parity error at the ADU.

Subsection 43 (SUB43)

This subsection tests the force D5 full at the ADU.

Subsection 44 (SUB44)

This subsection tests the force CMC busy at the CMI.

Subsection 45 (SUB45)

This subsection tests for OS bounds violation when slave PP tries to write CM outside allowed region specified by OS bounds.

Subsection 46 (SUB46)

This subsection tests the set inhibit PP CM request.

Subsection 47 (SUB47)

This subsection tests the clear inhibit PP CM request.

1

DST4 – Display Alignment Test

The Display Alignment Test (DST4) tests the interface to and from the CC545 display console. This test requires human interaction and all faults are detected by the operator.

Section Descriptions

DST4 is divided into eight sections.

Section	Tag	Description
00	SEC00	Tests dot function, CDC display code.
01	SEC01	Tests one character, full screen, CDC display code.
02	SEC02	Tests full alphabet, CDC display code.
03	SEC03	Tests intensity, CDC display code.
04	SEC04	Tests dot function, ASCII display code.
05	SEC05	Tests one character, full screen, ASCII display code.
06	SEC06	Tests full alphabet, ASCII display code.
07	SEC07	Tests intensity, ASCII display code.

A keyboard check may be accomplished by inputting characters to CMSE without pressing **CR**. These characters are displayed by CMSE, but are not interpreted as commands, unless the **CR** key is pressed.

Every subsection in DST4 has the following three conditions.

Condition	Description
00	Displays data on left screen.
01	Displays data on right screen.
02	Displays data on both screens.

Sections 00 and 04 (SEC00, SEC04)

These sections test the Dot function with CDC display code and ASCII display code, respectively. They have the following two subsections.

Subsection	Tag	Description
00	DOTTST	Displays 32-by-32 dot raster.
01	CRSTST	Displays crossed-diagonal lines.

Sections 01 and 05 (SEC01, SEC05)

These sections test the display of a full screen by filling the screen with one character. The operator enters the desired character when the message, ENTER CHARACTER is displayed. Each subsection contains the following three subsections.

Subsection	Tag	Description
00	DISCHR	Displays small characters.
01	DISCHR	Displays medium characters.
02	DISCHR	Displays large characters.

Sections 02 and 06 (SEC02, SEC06)

These sections display the full alphabet in CDC display code and ASCII display code, respectively. They have the following three subsections.

Subsection	Tag	Description
00	DISFUL	Displays small characters.
01	DISFUL	Displays medium characters.
02	DISFUL	Displays large characters.

Sections 03 and 07 (SEC03, SEC07)

These sections display the message INTENSITY ADJUST in small, medium, and large characters. The operator should attempt to adjust the intensity of the display station. These sections have only one subsection, INTENS.

CRA4 - Clock and Remote Access Test

The Clock and Remote Access Test (CRA4) tests the special features available on the two-port mux.

NOTE

CRA4 runs in standalone mode only (not under CMSE).

Section Descriptions

CRA4 is divided into six sections.

CRA4 does not require a terminal to be connected to the ports for sections 00 through 03. Sections 04 and 05 require that a pair of modems and a terminal be connected to the port to be tested. Refer to the paragraph titled Procedures for Remote Sections of CRA4 for information about connecting the modems. The test uses even parity and two-stop bits for sections 04 and 05. These settings cannot be changed. TMP4 must run error free before CRA4 can run.

NOTE

CRA4 requires operator intervention to run sections 04 and 05.

Section	Tag	Description
00	SEC00	Reads and displays the wall clock.
01	SEC01	Tests the wall clock by writing different test patterns.
02	SEC02	Tests the wall clock increment feature.
03	SEC03	Enter the time in the wall clock.
04	SEC04	Tests auto-answer on port 0.
05	SEC05	Tests auto-answer on port 1.

Procedures for Remote Sections of CRA4

If either or both sections 04 and 05 are selected, CRA4 displays the following message after section 03.

```
AUTO ANSWER TEST
-   SET PORT-OPTION SWITCH TO
    MSG ONLY OR DS-POWER ENABLED
-   CONNECT MODEM ON
    PORT 0
-   SPACE BAR TO CONTINUE
```

Sections 04 and 05 can only be run from a remote terminal (separate from the operator console) over a telephone link as shown in figure 1-1.

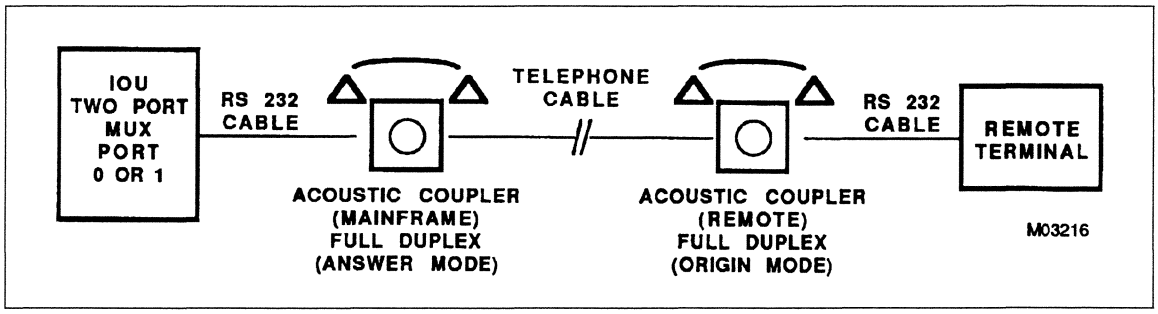


Figure 1-1. Auto-Answer Test Connections

The following procedure is used to run section 04.

1. Connect modem to the system port (port 0 for section 04; port 1 for section 05).
2. Connect the remote terminal to a modem.
3. Set the port-option switch on the two-port-mux box to **MSG ONLY**.
4. Press the **Space Bar** on the operator console to start the test. CRA4 monitors the ring indicator signal by reading the port status after you press the **Space Bar**. After 1 minute, the section is aborted.
5. Dial the telephone number of the system from the telephone connected to the remote terminal. When CRA4 detects two rings, it sets the Data Terminal Ready (DTR) and Request To Send (RTS) signals and sends the following message to the remote terminal. There is a delay of 15 seconds before this message is displayed.

   ```
   ENTER CHARACTER-UPPER CASE
   OR
   SPACE BAR TO END SECTION
   ```

   If a character is entered, it is echoed back to the remote terminal.
6. Press the **Space Bar** on the remote terminal to complete the section and advance to the next section.

   If section 05 is selected, CRA4 displays the following message after section 04.

   ```
   AUTO ANSWER TEST
   - SET PORT-OPTION SWITCH TO
     MSG ONLY OR DS-POWER ENABLED
   - CONNECT MODEM ON
     PORT 1
   - SPACE BAR TO CONTINUE
   ```

   Repeat steps 1 through 6 to run section 05.

TPM4 – Two-Port Multiplexer Test

The Two-Port Multiplexer Test (TPM4) tests the interface to and from the connected terminals. The first four sections run independently and require no operator interaction, except at error stops. All other sections (05 through 08) of the test require operator interaction, and the faults detected by the test are reported on the deadstart terminal.

The test exercises only the nondeadstart ports. At least one test terminal must be connected to execute the test. If the CC634-B is connected to port 0, it tests only port 0. To test the other port, the CC634-B must be connected to the other port, or two terminals can be used, one connected to each port.

The test runs with the CC634-B set in page mode and with cursor positioning enabled (switch inside CC634-B).

The following message is displayed only at the beginning of the test when the set PARAM message is displayed.

```
SET DISPLAY TERMINAL

ON LINE,  PAGE MODE,  FULL DUP.

(LOC 141B)=PARITY SELECT

00=ODD,  01=EVEN,  02=NO PARITY

(LOC 142B)=STOP BITS

00=1 STOP BIT,  01=2  STOP BITS.

DEFAULT-EVEN PARITY/2 STOP BITS
```

When parameter word 142 has the upper bit set (1xxxxx), the two-port-mux test expects one of the ports to be connected to a modem, and the following message is displayed after section 04 has completed execution.

```
TPM4 FOUND NO CARRIER ON STATUS
THE MODEM PARAMETER IS SELECTED.
DIAL PHONE FOR CONNECT TO TWO PORT MUX
TEST WILL WAIT FOR CARRIER ON STATUS
```

Section Descriptions

TPM4 is divided into the following nine sections.

Section	Tag	Description
00	SEC00	Tests function response and FIFO.
01	SEC01	Tests the loopback with 5-, 6-, 7-, and 8-bit mode (8-bit only when connected to a PC console).
02	SEC02	Tests data in overrun.

Section	Tag	Description
03	SEC03	Tests the DTR function and Connect function.
04	SEC04	Tests Master Clear and Disconnect functions.
05	SEC05	Tests X-Y positioning, displays dot code, and parity-error detection.
06	SEC06	Tests display full alphabet.
07	SEC07	Tests X-Y positioning by displaying cross-diagonal lines.
08	SEC08	Tests the read deadstart port, Terminal function.

Section 00 (SEC00)

This section has the following four subsections.

Subsection	Tag	Description
00	SUBSEC00	Checks to see if the test initialization detected a response to the DTR function.
01	SUBSEC01	Checks for the response on issued functions.
02	SUBSEC02	Checks for FIFO operation.
03	SUBSEC03	Checks if the DSR signal remains set if previously set by the DTR function after the FIFO has emptied.

Section 01 (SEC01)

This section has the following four subsections.

Subsection	Tag	Description
00	SUBSEC00	Tests the loopback feature with 8-bit mode.
01	SUBSEC01	Tests the loopback feature with 7-bit mode.
02	SUBSEC02	Tests the loopback feature with 6-bit mode.
03	SUBSEC03	Tests the loopback feature with 5-bit mode.

NOTE

When the deadstart console is a PC type, the test defaults to only testing 8-bit mode, because of the default hardware in the two port mux.

Section 02 (SEC02)

This section tests the data in overrun bit.

Each subsection has two conditions. Condition 00 tests port 0 (if connected) and condition 01 tests port 1 (if connected).

Section 03 (SEC03)

This section has the following four subsections.

Subsection	Tag	Description
00	SUBSEC00	Tests the Data Terminal Ready function (40X).
01	SUBSEC01	Same as subsection 00.
02	SUBSEC02	Tests the Connect function.
03	SUBSEC03	Verifies that the TPM will not clear the Data Terminal Ready signal after the FIF0 has emptied if it was set previously by the Set-Data-Terminal-Ready function.

Section 04 (SEC04)

This section has the following two subsections.

Subsection	Tag	Description
00	SUBSEC00	Tests the Master-Clear function (700) and verifies that the input and output buffers are cleared with this function.
01	SUBSEC01	Tests the Disconnect function (6000). Checks that the two-port mux will continue to output after the disconnect when data is still in the buffer.

Section 05 (SEC05)

This section has the following two subsections.

Subsection	Tag	Description
00	SUBSEC00	This subsection displays the message, ENTER CHARACTER. The test checks to see if characters can be entered while the two port mux is outputting.
01	SUBSEC01	Tests the parity-error detection and displays a full screen of the character. The character is entered by the operator when the ENTER CHARACTER message is displayed.

Section 06 (SEC06)

This section displays the full alphabet.

Section 07 (SEC07)

This section tests X-Y positioning using cross-diagonal lines.

Section 08 (SEC08)

This section tests the Read Deadstart Port and Terminal function. The deadstart port and terminal are displayed on the console and operater input is required to continue the test.

1

TPA4 – Two-Port Multiplexer Test (Standalone)

This Two-Port Multiplexer Test (TPA4) is a standalone only test.

TPA4 tests the interface between the PP using channel 15_8 and port 0 or 1. The test requires user interaction and all faults are detected by the operator when the test stops with a display message error or a hung condition. The test stops in a hung state only if a function code failure occurs.

The test exercises ports 0 and 1 when a CC545 is used as the main console. Only port 1 is exercised when the PC is the main console.

The read and write clock sections are run from both ports and therefore test all ports.

NOTE

If sections 07 and 08 are run it is necessary to reset the wall clock before the operating system is loaded because of a mismatch between the two-port-mux clock and the last entry made in the PC console clock monitor. This mismatch may cause software update problems.

The test is run with the quicklook bit cleared so that the operator can see the subsection and condition counters incrementing. Otherwise, too much time is wasted determining the state of the PP, running or not running.

The following message is displayed only at the beginning of the test when the set PARAM message is displayed.

```
SET DISPLAY TERMINAL

ON LINE,  PAGE MODE,  FULL DUP.

(LOC 141B)=PARITY SELECT

00=ODD,  01=EVEN,  02=NO PARITY

(LOC 142B)=STOP BITS

00=1 STOP BIT,  01=2  STOP BITS.

DEFAULT-EVEN PARITY/2 STOP BITS
```

Section Descriptions

The TPA4 test is divided into the following ten sections.

Section	Tag	Description
00	SEC00	Tests the Read Deadstart Port and Terminal function.
01	SEC01	Tests the function response and FIFO.
02	SEC02	Tests the DTR and Connect function.
03	SEC03	Tests the loopback with 5-, 6-, 7-, and 8-bit mode.
04	SEC04	Tests the Master Clear and Disconnect functions.
05	SEC05	Tests the data in overrun.
06	SEC06	Reads and displays the wall clock.
07	SEC07	Tests the wall clock by writing different test patterns.
08	SEC08	Tests the wall clock increment feature.
09	SEC09	Enters the time in the wall clock.

Section 00 (SEC00)

This section tests the Read Deadstart Port and Terminal function. The deadstart port and terminal are displayed on the console and operator input is required to continue the test. If the CC545 is the monitor console, the Read Deadstart function is executed on both ports and displayed.

Section 01 (SEC01)

This section has the following four subsections.

Subsection	Tag	Description
00	SUBSEC00	Checks to see if the test initialization detected a response to the DTR function.
01	SUBSEC01	Checks for the response on issued functions.
02	SUBSEC02	Checks for FIFO operation.
03	SUBSEC03	Checks if the DIR signal remains set if previously set by the DTR function after the FIFO has emptied.

Section 02 (SEC02)

This section has the following four subsections.

Subsection	Tag	Description
00	SUBSEC00	Tests the Data Terminal Ready function (40X).
01	SUBSEC01	Same as above.
02	SUBSEC02	Tests the Connect functions.
03	SUBSEC03	Verifies that the TPM will not clear data terminalready signal after the FIFO has emptied if it was set previously by the Set Data Terminal Ready function.

Section 03 (SEC03)

This section has the following four subsections.

Subsection	Tag	Description
00	SUBSEC00	Tests the loopback feature with 8-bit mode.
01	SUBSEC01	Tests the loopback feature with 7-bit mode.
02	SUBSEC02	Tests the loopback feature with 6-bit mode.
03	SUBSEC03	Tests the loopback feature with 5-bit mode.

Section 04 (SEC04)

This section has the following two subsections.

Subsection	Tag	Description
00	SUBSEC00	Tests the Master Clear function (700). This subsection will display the message `Enter character`. The operator enters the character that is required to fill the input buffer to test the Master Clear function.
01	SUBSEC01	Tests the Disconnect function (6000).

Section 05 (SEC05)

This section tests the data in overrun bit.

Each subsection has two conditions. Condition 00 tests port 0 and condition 01 tests port 1. The testing is accomplished through the use of loopback and input data check.

Section 06 (SEC06)

This section has one subsection. The subsection reads the wall clock and displays it on screen. Each condition is a different read.

Section 07 (SEC07)

This section tests the wall clock by writing a pattern of data to clock then reading it back to verify. There are 5 fields defining the year (YY), month (MM), day (DD), hour (HH), and minute (MN) as shown in the following example.

```
:---------------------------------:
: 00 : YY : MM : DD : HH : MN : 00 :
:    :    :    :    :    :    :    :
```

Field	Condition	Patterns
YY	0	00
YY	1	01
.	.	.
.	.	.
YY	98	98
YY	99	99
MM	100	01
MM	101	02
.	.	.
.	.	.
MM	110	11
MM	111	12
DD	112	01
DD	113	02
.	.	.
.	.	.
DD	141	30
DD	142	31
HH	143	00
HH	144	01
.	.	.
.	.	.
HH	165	22
HH	166	23
MN	167	00
MN	168	01
.	.	.
.	.	.
MN	225	58
MN	226	59

In each condition, a pattern is written in a field; other fields contain 01. For example:

Condition 03: 00 03 01 01 01 01 00
Condition 98: 00 98 01 01 01 01 00
Condition 108: 00 01 09 01 01 01 00.

1

Section 08 (SEC08)

The section has the following three subsections.

Subsection	Tag	Description
00	SUBZER	This subsection checks propagation delay from minute to hour to day to month and to year.

NOTE

To run subsections 1 and 2, clear quicklook mode bit.

01	SUBONE	This subsection checks the propagation of the carry (the increment of the clock). minute → hour, hour → day, day → month, month → year.
02	SUBTWO	This subsection tests the calendar clock to make sure that the last day is right for each month.

Section 09 (SEC09)

The section has the following subsection.

Subsection	Tag	Description
00	SEC090	This subsection can be used to set date and time in the wall clock from the keyboard. When the message ENTER DATE/TIME YYMMDDHHMN is displayed, enter data through the keyboard in the above form, where YY = year, MM = month, DD = day, HH = hour, MN = min. No separators are required. For example: 8506100812. If data is illegal an INVALID ENTRY message is displayed. If good data is entered, it is written to the clock, read back and displayed. Type ABS **CR** to exit or abort this subsection at the end of the test.

Isolation Diagnostics

The isolation diagnostic (FII44) analyzes errors reported by the detection tests and attempts to identify a group of logic paks that are likely to be causing the problem.

FII4 – Fault Isolation Program

The Fault Isolation Program (FII4) is a table-driven analyzer that uses data recorded by the IOUCP Error Processing Routine (ERRCP) during IOU test execution.

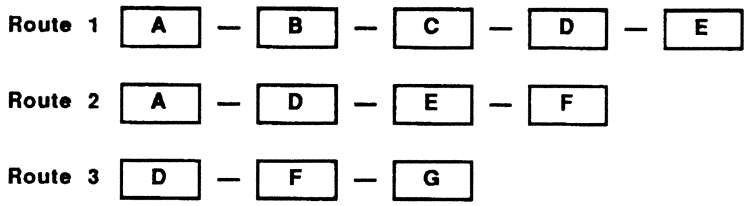
ERRCP records the following three classes of errors and assigns route numbers to them.

- Unexpected parity errors - (route 0)
- Invert parity errors - (routes 1 through 77_8)
- Data/control errors - (routes 100 through 177_8)

FII4 analyzes the errors recorded by the detection tests using the route numbers as the base for isolation.

The route numbers for fault isolation are an index to a predefined string of modules.

Example:



FII4 sets error location (ERROR) to zero and starts by processing route numbers 100 through 177_8. If an error is found in these route numbers, it sets error location (ERROR) to nonzero. If no errors are found for the routes, FII4 processes route 0. If no error is found for route 0, FII4 processes errors for routes 1 through 77_8.

When FII4 finds errors in the route numbers, it calls subroutines to find and identify the pak associated with the highest number of error routes. The subroutine assigns the priority number one to paks with the highest count and then identifies lower priority paks. The maximum number of paks that can be called is 10.

Log Errors Routine

This routine is built into the IOUCP. It is used to log errors in the error buffer for fault isolation. It is called by ERRCP. The following classes of errors are logged.

- Unexpected parity error: When a parity error is detected, IOUCP generates a ROUTE=0 code and passes the contents of Fault Status registers 1 and 2 (FS1 and FS2) in (PEBUF) to the Error Log Routine.

- Invert parity error: When MRT4 detects an error, it generates a ROUTE=1 through 77_8 code and passes the logical difference between expected and received FS1 and FS2 in PEBUF to the Error Log Routine.

- Data/control error: When tests detect an error, they generate a ROUTE=100 through 177_8 code and passes the following parameters to the Error Log Routine.

(EXPAT)	Expected Pattern Buffer
(RCPAT)	Received Pattern Buffer
(PPNUM)	Failing PP Number
(CHNUM)	Failing CH Number
(PEBUF)	Parity Error Buffer

Figure 1-2 shows the Monitor PP map. The errors are recorded by the Error Log Routine in the Error Log Buffer and processed by FII4.

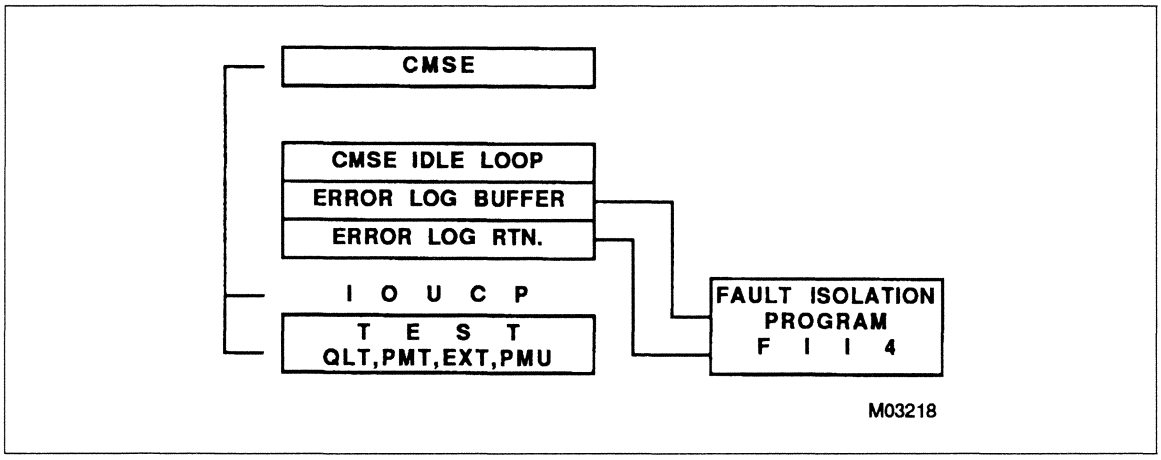


Figure 1-2. Monitor PP Map

Unexpected Parity Error Recording

The route number=0 is transmitted whenever unexpected parity errors occur. As shown in figure 1-3, the errors are logged by setting the RPEBUF equal to the accumulative logical difference of FS1 and FS2 from PEBUF, and PCOUNTER is incremented by one.

The size of the recorded parity error buffer (RPEBUF) is 20_8 locations.

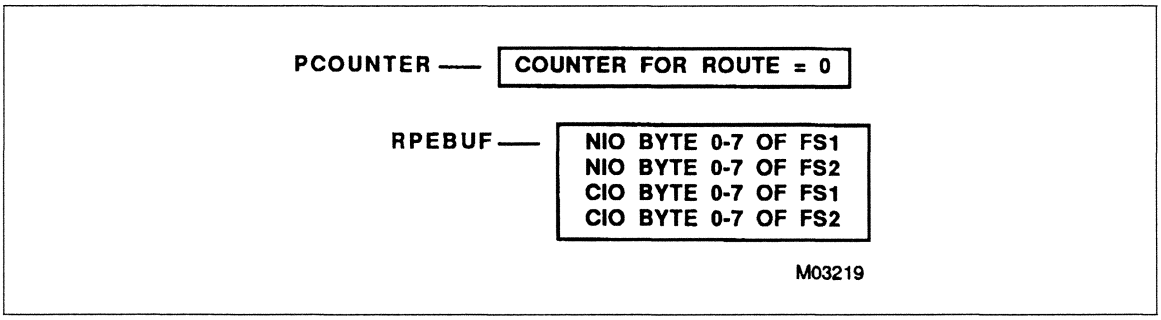


Figure 1-3. Parity Error Buffer

Invert Parity Error Recording

The routes numbered 1 through 77_8 are invert parity error routes. When one of these errors is reported, the corresponding location (PCOUNTER+(ROUTE)) is incremented by one.

Also, a cumulative logical difference of FS1 and FS2 is recorded in location RPEBUF+20_8 to location RPEBUF+37_8.

Figure 1-4 shows what these locations represent.

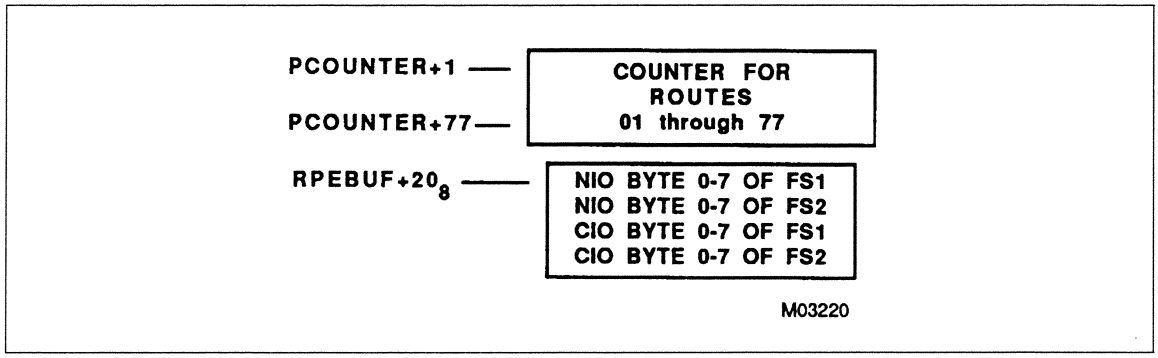


Figure 1-4. Invert Parity Error Buffer

Control and Data Error Recording

The routes numbered 100 through 177_8 represent the errors for data and control flow. The route numbers are used to generate the starting address for cumulative logical difference, PP, and channel flags when an error is detected.

The logical difference is calculated using expected (EXPAT) and received (RECPAT) patterns. The PP and channel position flags are calculated using (PPNUM) and (CHNUM).

The corresponding counter for failing route (PCOUNTER+(ROUTE)) is incremented by one.

Figure 1-5 shows the buffer for one failing route number.

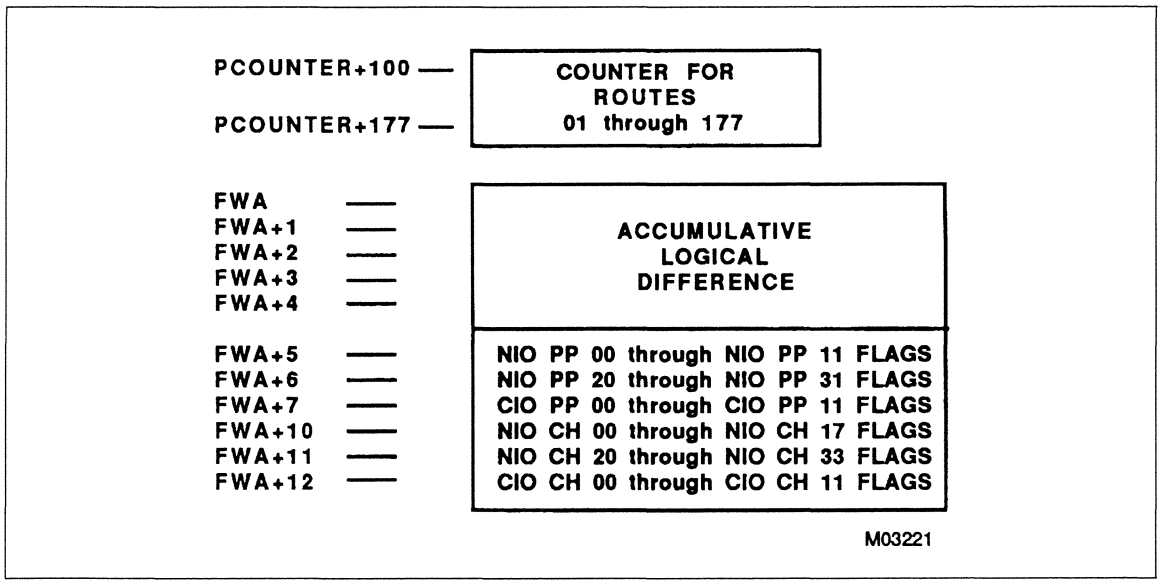


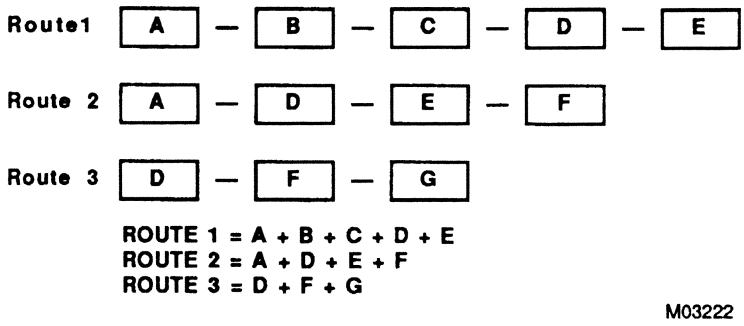
Figure 1-5. Control and Data Error

Fault Isolation Program

FII4 analyzes the errors recorded by the detection tests using route numbers as a base for isolation.

The route numbers for FII4 are considered as a index for the string of the involved modules.

Example:



FII4 sets location (ERROR) to zero; initialize no error condition.

It starts with the processing of the route numbers 100 through 177_8. These route numbers represent the data and control flow errors. This processing is done by checking the corresponding location in the counter buffers for nonzero.

When a nonzero is found, location (ERROR) is set to nonzero.

If no error for these routes was found, it processes the unexpected parity error route (ROUTE=0). If the counter for this route is 0, it processes the invert parity routes (1 through 77_8). If there are no errors for these routes, it sends the message to CMSE: NO ERROR RECORDED.

If location (ERROR) is nonzero, Fault Isolation Control Program (FICP) calls FISORT routine to find the pak with the highest number of calls. The counter for that pak has the highest count.

The FISOL routine is then called to set priority and identify the pak type and location. The calling table is passed to CMSE to display the message SUSPECTED FAILING MODULE/S (FISM1, Fault Isolation Message 1).

Data and Control Processing

The errors with route numbers 100 through 177_8 are processed in the following manner:

- Use COPYER routine to set the following:
  - Accumulative logical difference to received buffer.
  - Failing PP flags in PP parameter area.
  - Failing channel flags in channel parameter area.
  - Number of failing barrels (NUMBAR).
  - Failing physical barrel number (BARNUM).
- Call the processor for that route number to:
  - Establish if it is a data or control failure.
  - Find the number of failing bits and failing bit numbers.
  - Find the failing channel number and number of failing channels if it is channel failure.
  - Increment counters for the paks that cause the error.

Invert Parity Processing

This processor checks if any location in the PCOUNTER buffer is nonzero and, if not, returns control to Fault Isolation Control Program (FICP) with location (ERROR) unchanged. It sets location (ERROR) to nonzero if any failing route is found. If there is any error, it calls COPYPE routine to copy the recorded accumulative logical difference to PEBUF and:

- Sets NUMBAR.
- Sets BARNUM.

The processor then increments counters for involved paks by one.

Parity Error Processing

This processor checks if location PCOUNTER is nonzero and, if not, returns control to FICP with location (ERROR) unchanged. If there is, error location (ERROR) is set to nonzero and the COPYPE routine is called to copy the recorded accumulative logical difference for this route in PEBUF and:

- Sets NUMBAR.
- Sets BARNUM.

The parity error processor then increments the involved paks by one.

Module Sorting

The Pak Sorting Routine (FISORT) is used to find the pak counter with the highest count and set Maximum Count Value (MAXCNT).

If the highest count is zero (processors were not able to determine failing pak multiple uncorrelated error), the FISM2 (Fault Isolation Message 2) is passed to CMSE.

Fault Isolation

The Fault Isolation Routine (FISOL) is called to identify the pak type and pak location and to set the priority for the paks that have counters nonzero.

FISOL assigns the priority number one to all the paks with the highest counters. It then decrements maximum count and increments priority number for the next set of paks with the same count as current maximum count.

FISOL calls MODID for pak identification and location.

Maximum number of paks to be called is 10.

AT511-A/AT512-A IOU and Associated Equipment 2

AT511-A/AT512-A IOU and Associated Equipment 2

LDX – Long Deadstart Sequence

The Long Deadstart Sequence (LDX) is a quick look program that resides in read-only memory (ROM) attached to upper core of logical PP00 memory (5400 through 7777_8). LDX transfers to barrel 1 after barrel 0 completes. LDX continues to transfer to barrels 2 and 3 if they are installed. Peripheral processors (PPs) within each barrel may be reconfigured by using the deadstart display and entering the RP x command. Initiating the long deadstart sequence sets the following conditions.

- A master clear is issued to all PPs and all error status bits are cleared.
- PP00 is forced to read the test program stored in ROM.
- PP00 address is forced to 6000_8 (5400 through 5700_8 is for test update and transfer routine).
- A PP register display is brought up on the right-hand side of the screen.

While LDX is executing, certain error status bits are monitored to determine if the hardware has been detected. If a hardware failure is detected, a message is displayed below the PP register.

Each error stop is indicated by a unique P register address. The customer engineer (CE) determines what error condition was detected by examining the P register display and referring to the LDX error tables. Refer to the CYBER 960 Series and 970 Series System Troubleshooting Guide, listed in the preface of this manual.

When execution of LDX is finished, (all installed barrels run), the short deadstart sequence is initiated using the program currently displayed on the screen.

LDX Tests

LDX performs the following sequence of tests:

1. Arithmetic Unit Test
2. PP00 Memory Test (up to location 5377_8)
3. One Word Channel Test
4. Block I/O Transfer Test
5. Block I/O Transfer with Conversion Test
6. Multi-PP Arithmetic Unit Test
7. LDX Bit Test
8. Detection Tests

The following paragraphs describe these tests.

Arithmetic Unit Test

The tests performed by the Arithmetic Unit Test are:

Test	Description
Unconditional Jump Test	Checks execution of the 03 instruction and the lower 6 bits of the d portion of the instruction. Error stops are: – on 0300 or 0377 instruction for single Q-bit failure. – indicated by hardware timeout counter error for a multiple Q-bit failure or P register failure.
Conditional Jump Test	Checks instructions 04 through 07 with A equal to 0, A equal to 1, and A equal to minus 0. Error stops are on 03 through 07 instructions.
1x Instructions Test	Checks instructions 10 through 17 using the lower 6 bits of the A register. Error stops are on branch instructions.
2x Instructions Test	Checks instructions 20 through 25, arithmetic operations in the A register with all 18 bits, and shift network. Error stops are on branch instructions.
3x Instructions Test	Checks instructions 30 through 37 using the lower addresses of PP00 memory for store instructions. Error stops are on branch instructions.
4x Instructions Test	Checks instructions 40 through 47 and all 16 bits of the Q register. Lower addresses of PP00 memory are used to check restore capability. Error stops are on branch instructions.
5x Instructions Test	Checks instructions 50 through 57 with and without the Q flag set. Error stops are on branch instructions.
13x Instructions Test	Checks execution of instructions 130 through 137 with 16-bit operands. Error stops on branch instructions.
14x Instructions Test	Checks execution of instructions 140 through 147 with 16-bit operands. Error stops are on branch instructions.
15x Instructions Test	Checks execution of instructions 150 through 157 with 16-bit operands, with and without the Q flag set. Error stops are on branch instructions.
RJM and LJM Instructions	Checks execution of instructions. Test 01 and 02 using previously written addresses in lower core of PP00 memory. Error stops are on branch instructions.

PP00 Memory Test

This test executes from low addresses of PP memory. The test is transferred from ROMs to PP memory and checked for correct transfer. Error stop is on a branch instruction (PP00 at location 24_8, PP01 through PP04 at PP00 memory location 210_8 plus PP number). The test then checks PP memory with ten fixed data patterns and

addresses. Error stop is on a branch instruction. The contents of the A register represent the failing bits. The failing pattern and failing address are stored in low addresses of PP00 memory. The following table indicates the bit locations in PP00 memory.

PPM00 Location	Contents
34_8	Failing bits
35_8	Failing patterns
36_8	Failing addresses

One Word Channel Test

This test checks channels with one word input/output (I/O) using all possible bit combinations and Channel status during I/O. Barrel 0 tests channels 0, 1, and 13. All other barrels test channels 12, 15, and 17. The remaining channels are not tested because they may not be installed. Error stops are on branch instructions and are unique for each Channel status and data error.

Block I/O Transfer Test

This test checks inter-PP communication channel 12 using a slightly modified PP memory test. A timeout counter protects PP00 from a hung condition. Error stops are on branch instructions that are unique for each channel/PP.

Block I/O Transfer with Conversion Test

This test checks inter-PP communication with 12-to-16-bit and 16-to-12-bit conversion over channel 12. Error stops are on branch instructions. A timeout feature protects PP00 from a hung condition.

Multi-PP Arithmetic Unit Test

This test is the same as the Arithmetic Unit Test described previously, but runs in PP01 through PP04 simultaneously. If an error is detected, the PP in which the test is running stops and P register stop lights are the same as for the Arithmetic Unit Test. PP00 is in a loop waiting for a message from PPxx indicating that the test successfully passed. The looping address is unique to each PP. Each PP is force deadstarted, waits for the channel 13 flag to set before starting, and sets the channel 12 flag when completed.

LDX Bit Test

This test checks the LDX bit in the Maintenance register. If the bit is clear, PP00 stops and the P register indicates an error in the Maintenance register.

Detection Tests

The detection tests are a series of programs that test each PP and channel in the IOU4C for correct functional operation. These tests also record errors such as differences between expected and actual (received) results and generate internal codes that identify the area of the IOU4C under test when the error was detected.

QLT44 – Quick Look Test

The Quick Look Test tests one-word I/O over each channel. The test uses hardware features such as force idle PP and force deadstart PP.

At the end of the test, all tested channels are set active, empty, and all associated PPs, if selected, are in an idle condition.

Section Descriptions

The Quick Look Test section numbers, converted to octal, represent the channel under test (that is, section 8 represents channel 10_8). For a channel to be tested, both the section and the channel must be selected by bits in the section select and channel flag parameter words. Refer to parameters and control words in the MSL 153/155 IOU Test Procedures Reference manual, listed in About This Manual.

This section has the following four subsections.

Subsection	Tag	Description
00	SUBSEC00	Initializes the associated PP (Force idle PP).
01	SUBSEC01	Tests channel flags status for set and clear.
02	SUBSEC02	Tests one word pattern over each channel selected.
03	SUBSEC03	Checks negative testing.

At the end of the section, channel status is active, empty, and the associated PP is in an idle condition.

Subsection 00 (SUBSEC00)

Initialize channel number for all instruction in the slave PP. Deadstart PP and check K register for 107700 idle condition.

Subsection 01 (SUBSEC01)

Test all channel flags for set/clear.

Subsection 02 (SUBSEC02)

Subsection 02 tests one word pattern over each channel selected to be tested, and checks the status of the channel during the I/O transfer. This subsection has 1000_8 conditions that use different patterns and complements of the pattern. There are 152_8 fixed patterns and 626_8 randomly generated patterns.

Each condition uses one pattern and the complement of the pattern. During this I/O, the test checks for correct channel status and channel error flag clear status.

Subsection 03 (SUBSEC03)

Subsection 03 has the following ten conditions.

Condition	Description
00	Disconnects already inactive on channel with escape bit set.
01	Outputs one word on inactive channel with escape bit set.
02	Inputs one word from inactive channel with escape bit set.
03	Blocks input over inactive channel expecting that the first word address of the block input is cleared.
04	Blocks output over inactive channel expecting that the contents of the A register are not changed.
05	Activates already active channel with escape bit set.
06	Functions FAN on active channel with escape bit set.
07	Functions FNC on active channel with escape bit set.
08	Forces channel active, full, and check for empty when channel is deactivated.
09	Checks channel for empty and active signals.

NOTE

It is possible for subsection 03 to cause the PP from which the test is being executed to hang if the escape feature (bit 2^5 of I/O instructions) does not work. There is no way to prevent this hang if the feature fails.

If all channels for the current barrel are turned off, channel 17_8 is used as a default test channel.

PMT44 - PP Memory Test 1

The PP Memory Test 1 (PMT44) checks the operation of the PP memories except for the PP in which the PMT44 test resides. The test does not rely on code executing in the PP memory being tested. PP memories are tested sequentially using force deadstart, idle, and force dump. Since the PP in which PMT44 resides was checked by the LDX test, PMT44 assumes that the PP is operational.

Section Descriptions

The PP Memory Test 1 section numbers, in decimal, represent the PP memory being tested. For a memory to be tested, both the section and the PP must be selected by bits in the section select and PP flag parameter words. Refer to parameters and control words in the MSL 153/155 IOU Test Procedures Reference manual listed in the preface of this manual.

All data transfers for PPs are done over channel 13_8.

Each section has 32 subsections numbered 00 through 31, as described in the following table.

Subsection	Tag	Description
00	B00.00	Forces deadstart PP with check.
01	B01.00	Checks P and A registers.
02 to 30	B02.00	Tests memory with patterns.
31	B03.00	Tests memory with address patterns.

At the beginning of the test, all PPs are forced into a hung condition by a force idle feature. The PPs are left idle at the end of each section and are in an idle condition at the end of the test.

Subsection 00 (B00.00)

This subsection has the following five conditions.

Condition	Description
00	Forces PP to deadstart condition.
01	Checks P register of the PP being tested for the value 177777_8.
02	Checks Q register of the PP being tested for deadstart channel value.
03	Checks K register of the PP being tested for the value 007100_8.
04	Checks A register of the PP being tested for the value 020000_8.

Subsection 01 (B01.00)

This subsection has the following condition.

Condition	Description
00	Repeat one word output to fill tested memory. Check the P and A registers of the PP being tested for the correct count after each word of output.

Subsections 02 through 15 (B02.00)

These subsections output standard test patterns to complement every second word and then dumps the test PP over the used channel to the monitor PP to check the data.

These subsections have the following two conditions.

Condition	Description
00	Outputs data blocks of 100_8 words to fill up the test PP memory.
01	Forces a dump of the test PP and inputs one word at a time,checking data.

Subsections 16 through 30 (B02.00)

These subsections use the same code as subsections 02 through 15, but use random number patterns.

Subsection 31 (B03.00)

This subsection outputs the address of the corresponding memory location to fill the test PP memory and then dumps the test PP over the used channel to the monitor PP to check the data.

This subsection has the following two conditions.

Condition	Description
00	Repeats one word output to fill the memory of the test PP.
01	Forces a dump of the test PP and inputs one word at a time, checking data.

EXT44 - Execution Unit Test

The Execution Unit Test checks instruction execution and the arithmetic units in all PPs, except the one in which the Input/Output Control Program (IOUCP) and CMSE are running. CMSE loads EXT44 into the same PP with IOUCP and distributes the updated copies to the PPs to be tested. The test is started by deactivating the communication channel for the assigned PP.

Section Descriptions

EXT44 is divided into the following three sections.

Section	Tag	Description
00	UJNTST	Tests the arithmetic unit and execution of instructions 00 through 57_8 (12-bit operand instructions).
01	TEST13X	Tests the arithmetic unit and execution of instructions 1030 through 1057_8 (16-bit operand instructions).
02	STSTEST	Tests the execution of the channel instructions.

Section 00 (UJNTST)

This section tests the 12-bit operand instructions 00 through 57_8. The test is divided into the following six subsections.

Subsection	Tag	Description
00	UJNTST	Tests execution of 00 through 07 instructions.
01	TEST1X0	Tests execution of 10 through 17 instructions, and the A register bits 2^0 through 2^5.
02	TEST2X0	Tests execution of 20 through 23 instructions, and the A register bits 2^0 through 2^{17}.
03	TEST3X	Tests execution of 30 through 37 instructions, and restore ability.
04	TEST4X	Tests execution of 40 through 47 instructions, and the A register bits 2^0 through 2^{11}.
05	TEST5X	Tests execution of 50 through 57 instructions, and the Q adder.

Section 01 (TEST13X)

This section tests the 16-bit operand instructions. The test is divided into the following four subsections.

Subsection	Tag	Description
00	TEST13X	Tests execution of 1030 through 1037 instructions and restore ability on 16-bit boundary.
01	TEST14X	Tests execution of 1040 through 1047 instructions and restore ability using indirect addressing.
02	TEST15X	Tests execution of 1050 through 1057 instructions using index addressing.
03	JMPTEST	Tests execution of 01 and 02 instructions.

Section 02 (STSTEST)

This section tests the I/O instructions (channel instructions) and has the following four subsections.

Subsection	Tag	Description
00	STSTEST	Tests 64 through 67 instructions.
01	IANTEST	Tests 70, 72, 74, and 75 instructions.
02	IOTEST	Tests 1064 through 1067 instructions.
03	ESCTEST	Tests channel instructions with escape bit (Q bit 2^5 set).

PMU44 - PP Memory Test 2

The PP Memory Test 2 (PMU44) performs a more rigorous check of the PP memories than PMT44. The advantage over PMT44 is that PMU44 loads a copy of the PP memory test routine into all PP memories that are to be checked. The PP memory test routine is then executed by the PPs. This provides more variety in the way memory is referenced and allows more strenuous code to be used.

Section Descriptions

The section, subsection, and condition structure of PMU44 is defined in reference to where the code executes in a particular test PP. The code in the test PP is executable from either the lower or upper half of the first 4K of PP memory, so that the entire memory can be tested.

Selection of sections 00 and 01 tests the entire memory.

Section	Description
00	Executes in the lower half of memory and tests all memory except test core and direct cells used by the lower half copy.
01	Executes in the upper half of memory and tests direct cells and core used by the lower half copy.

Each section includes all of the following 48 subsections except as noted.

Subsection	Tag	Description
00	SS00	Checks to find a solid addressing failure. This is not affected by the memory bounds of a section.
01 through 36	SS01 through 36	Checks standard and random patterns of available test memory.
37 through 39	SS37 through 39	Tests available test memory with a random word in each location.
40 through 46	SS40 through 46	Checks addressing by rapidly loading random addresses in a pre-arranged sequence.
47	SS47	Checks if faults are induced when address lines are selectively toggled. This is not affected by the memory bounds of a section.

Subsection 00

This subsection uses the following two conditions to check for solid address failure.

Condition	Description
00	Initializes location 0 with all zeros, and stores ones in all remaining locations with one-bit-set addresses. A nonzero word in location 0 indicates a shorted address line.
01	Initializes location 7777_8 with all ones, and stores zeros in all remaining locations with one-bit-clear addresses. A zero-bit in location 7777_8 indicates an open address line.

Subsections 01 through 06

These subsections use a standard fixed pattern. The testable memory is checked in groups of five words with the pattern, then its complement, being loaded into alternate locations. The words are quickly read back and checked in sequential order.

Subsections 07 through 14

These subsections are the same as subsections 01 through 06, except the standard patterns are circularly shifted. These sliding patterns use 16 conditions, one for each circular shift.

Subsections 15 through 21

These subsections are exactly the same as subsections 01 through 06. The placement allows staggering of fixed and sliding pattern checks.

Subsections 22 through 36

These subsections are the same as subsections 01 through 06, except the patterns are randomly chosen before the tests are copied to the test memories.

Subsections 37 through 46

These subsections have one condition. Each subsection uses a real-time-clock input to generate a series of 1000_8 addresses. For each address the previous address is used as data. This daisy chain is then rapidly retraced using load instructions. If no error is generated by rapid changes of the address/data lines, the 1000th load is the reference point.

Repeating the condition uses the same seed for the random number generation; repeating the subsection uses a new seed for each cycle.

The subsections are duplicated to allow multiple cycles without selecting a repeat subsection.

Subsection 47

This subsection checks if the address lines are affected by crosstalk. An instruction sequence is executed that causes the address lines to switch from all zeros to all ones, except for one bit to be checked for staying zero. The sequence is repeated for all 13 address bits, and then another sequence is used for checking all ones toggling to all zeros (but one).

This subsection uses the following two conditions.

Condition	Description
00	Toggles all address lines, except the test bit, from zeros to ones by executing a store with indirect addressing through location 0 that contains a one-bit-clear address.
01	Toggles all address lines, except the test bit, from ones to zeros by executing an indexed long jump instruction, from location 7776_8 to location 0 plus a one-bit-set address.

CHD44 – Channel Test

The Channel Test (CHD44) checks inter-PP data transfer over each tested channel.

The test uses a pair of PPs for inter-PP transfer. One PP is called the PP transmitter and the other, the PP receiver. The PP transmitter communicates through the Double PP Driver (DPPD) with the Input/Output IOU Control Program (IOUCP) over the communicating channel. The PP receiver communicates with the PP transmitter over the testing channel.

The PP transmitter and the PP receiver are randomly selected at the beginning of each section and whenever the test is repeated.

The execution of the complete test with one channel is considered a section. The number of selected sections specifies the number of the channels to be tested for randomly selected PP pair (default is 32 sections selected).

When all selected sections are completed, the end of the test is reported and, if repeat test is selected, new pairs are randomly generated.

The maximum number of words (size of the I/O block) to be transferred is 1000_8.

Section Descriptions

In this test, a section is the execution of the complete program using one pair of PPs that are testing one channel. The PP receiver returns received data over the channel being tested to the PP transmitter. The PP-transmitter section is divided into the following five subsections.

Subsection	Tag	Description
00	SUBSEC00	Tests one word I/O over the channel being tested using the OAN instruction.
01	SUBSEC01	Tests one word I/O over the channel being tested using the OAM instruction.
02	SUBSEC02	Tests the channel flag operation (PP-receiver number is the channel number for the channel flag test).
03	SUBSEC03	Tests block I/O over the channel being tested using the OAM instruction.
04	SUBSEC04	Tests block I/O over the channel being tested using the OAPM instruction (conversion).

Subsection 00 (SUBSEC00)

Subsection 00 tests the data and Channel status during one word I/O using the OAN (single word output) instruction over the channel being tested. The data used for I/O are word counts. Every condition is represented by one word count.

Subsection 01 (SUBSEC01)

Subsection 01 tests the data and Channel status during one word I/O using the OAM (output block) instruction over the channel being tested. The data used for I/O are the data patterns used in subsections 03 and 04. Every condition represents one pattern.

Subsection 02 (SUBSEC02)

Subsection 02 cannot be deselected. If subsections 03 or 04 are selected, then subsection 02 is selected. It tests the operation of the channel flag (channel flag number is equal to the receiving PP number). This channel flag is used in subsections 03 and 04 to control the receiving PP.

Subsection 03 (SUBSEC03)

Subsection 03 tests the data and Channel status during block I/O using the output block (OAM) instruction over the channel being tested. The size of the block is represented by the word count. Every condition is one block size (one word count).

Subsection 04 (SUBSEC04)

Subsection 04 tests the data and Channel status during block I/O using the output block (OAPM) instruction with conversion over the channel being tested. The size of the block is represented by the word count. Every condition is one block size (one word count).

CMA44 - Central Memory Access Test

The Central Memory Access Test (CMA44) checks the data path from PP memory to central memory (CM) and from CM to PP memory. It also checks the Relocation (R) register which, in conjunction with the A register, forms the CM address.

Section Descriptions

CMA44 is divided into ten sections.

Section	Tag	Description
00	SEC00	Tests the R register.
01	SEC01	Tests the (R+A), CM address.
02	CWDTST	Tests the read/write CM with 60-bit single CM word.
03	ADRTST	Tests the read/write addresses in CM with 60-bit single CM word.
04	CWMTST	Tests the read/write CM with 64-bit single CM word.
05	CLDTST	Tests the read/write block in CM with 60-bit CM word.
06	CLMTST	Tests the read/write block in CM with 64-bit CM word.
07	RSLTST	Tests the read and set lock.
08	RCLTST	Tests the read and clear lock.
09	MIXTST	Tests the read/write in mixed mode (60- and 64-bit CM word, single CM word and block).

Section 00 (SEC00)

The section has the following three subsections.

Subsection	Tag	Description
00	RRG00	Tests the lower 12 bits of the R register, with the upper 10 bits of R set to zero.
01	RRG01	Tests the upper 10 bits of the R register, with the lower 12 bits of R set to zero.
02	RRG02	Tests the full R register.

Every subsection has 1000_8 conditions and every condition is a different pattern.

Section 01 (SEC01)

The section has the following subsection.

Subsection	Tag	Description
00	RAATST	Tests the full A+R register.

This subsection has 1000_8 conditions and every condition is a different pattern.

Sections 02 through 09

Sections 02 through 09 have 16 subsections each. Subsections 00 through 07 use different starting addresses in CM and do not use the R register. Subsections 08 through 15 use randomly selected R-register values.

Every subsection has 1000_8 conditions and every condition is a different pattern with a different starting address in CM.

Every condition is a different CM word count for the block read/write instruction.

MRA44 – Maintenance Register Access Test

The Maintenance Register Access Test (MRA44) checks the Maintenance register access from all PPs. It also checks the Maintenance Channel (MCH) hardware interlock and (MCH) priority circuits.

Section Descriptions

MRA44 has the following 26 sections.

Section	Tag	Description
00	SEC00	Tests the MCH interlock and PP priority.
01 through 25		Tests the MCH access from all PPs.

Section 00 (SEC00)

The section has the following two subsections.

Subsection	Tag	Description
00	S00B00	Tests that a PP from the lower barrel has priority over other PPs.
01	S00B01	Tests that only one PP can access the MCH at a time, using the MCH interlock.

Sections 01 through 25

Each section tests a MR access from the corresponding PP. Each section has the following five subsections.

Subsection	Tag	Description
00	FUNRES	Tests the response on function. Each condition is one function code.
01	ECHO	Tests the data to and from the MR using an echo function. Each condition is one pattern.
02	WRITE	Tests the Write function. The testing PP writes MR and the monitor PP reads and compares values.
03	READ	Tests the Read function. Each condition is one register in the MR.
04	SSUMF	Tests the Status-Summary function by reading data using the Status-Summary function and normal read operation.

MRT44 – Maintenance Register Test

The Maintenance Register Test (MRT44) checks all parity networks in the IOU4C using invert parity feature. All of the force conditions that induce errors to other parts of the hardware are also checked.

The 24-digit expected and received messages are the expected and received contents of the Fault Status 1 and Fault Status 2 registers.

Section Descriptions

The PP number is equal to the section number in test MRT44.

Subsection Descriptions

Each subsection is one of the Test mode register codes with 00 equal to Test mode code 00, 01 equal to Test mode code 01, and so forth, up to the code of 47 (decimal). Therefore, MRT44 is divided into the following 48 subsections.

Subsection	Tag	Description
00	SUB00	Tests for no reported errors.
01	SUB01	Tests for invert parity on channel data to PP.
02	SUB02	Tests for invert parity from PP to channel.
03	SUB03	Tests for invert PPM to R register parity.
04	SUB04	Tests for invert parity at PPM data checker.
05	SUB05	Tests for invert microcode parity.
06	SUB06	Tests for invert PPM parity at parity generator.
07	SUB07	Tests for invert parity on CM function code.
08	SUB08	Tests for invert parity on Y register at parity generator.
09	SUB09	Tests for invert parity on A register at parity generator.
10	SUB10	Tests for invert parity on shift ROM at parity checker.
11	SUB11	Tests for invert parity on Q register at parity generator.
12	SUB12	Tests for invert parity on P register at parity generator.
13	SUB13	Tests for invert parity on G register at parity generator.
14	SUB14	Tests for invert R register to Y parity at checker.

Subsection	Tag	Description
15	SUB15	Tests for invert PPM address parity at checker.
16	SUB16	Tests for invert tag-out parity at generator.
17	SUB17	Tests for invert data-in parity at generator.
18	SUB18	Tests for invert CM address parity.
19	SUB19	Tests for disable SECDED code to PP memory.
20	SUB20	Not used.
21	SUB21	Not used.
22	SUB22	Not used.
23	SUB23	Not used.
24	SUB24	Tests for force CM request/resync error.
25	SUB25	Tests for force mark line parity bit to zero.
26	SUB26	Tests for force tag-out/in parity error.
27	SUB27	Tests for force response code parity error.
28	SUB28	Tests for invert function-out parity bit.
29	SUB29	Tests for force address-out parity bit low.
30	SUB30	Tests for force data-in parity bit low.
31	SUB31	Tests for force data-out parity bit low.
32	SUB32	Tests for invert channel parity from the two-port mux or the maintenance channel.
33	SUB33	Tests for invert Maintenance register write parity.
34	SUB34	Tests for invert nanocode parity.
35	SUB35	Tests for invert read parity bit.
36	SUB36	Tests for invert channel 15 data-bus parity at checker.
37	SUB37	Tests for invert R/I read-data parity.
38	SUB38	Tests for invert R/I write-data parity.
39	SUB39	Tests for force lost clock error.
40	SUB40	Tests for force CM request in the ADU.
41	SUB41	Tests for force tag-in parity error in the ADU.

Subsection	Tag	Description
42	SUB42	Tests for force response code parity error in the ADU.
43	SUB43	Tests for block D5 full in the ADU.
44	SUB44	Tests for force CMC busy at CMI.
45	SUB45	Tests for invert OS bounds address parity at checker.
46	SUB46	Tests for set inhibit PP to CM request.
47	SUB47	Tests for clear inhibit PP to CM request.

NOTE

Subsections 46 and 47 must be run together because subsection 46 sets up the condition that subsection 47 clears. You cannot select one section without the other.

Subsection 00 (SUB00)

This subsection tests that no errors are reported when a code of (00) is forced in the Test Mode register.

Subsection 01 (SUB01)

This subsection tests for invert parity on a channel to PP data transfer.

Subsection 02 (SUB02)

This subsection tests for invert parity on a PP to channel data transfer.

Subsection 03 (SUB03)

This subsection is executed to cause parity error for PP to the R register.

Subsection 04 (SUB04)

This subsection is executed to cause parity error at the PP memory data checker.

Subsection 05 (SUB05)

This subsection causes parity errors in microcode and the G register.

Subsection 06 (SUB06)

This subsection tests the parity network for data sent to the PP memory.

Subsection 07 (SUB07)

This subsection tests the parity error on CM function.

Subsection 08 (SUB08)

This subsection tests the parity network of the Y register.

Subsection 09 (SUB09)

This subsection tests the A register parity error network.

Subsection 10 (SUB10)

This subsection shifts the data in the A register to cause a shift control ROM parity error.

Subsection 11 (SUB11)

This subsection shifts the data for checking parity in the Q register.

Subsection 12 (SUB12)

This subsection cause parity errors in the P register.

Subsection 13 (SUB13)

This subsection causes a parity error in microcode and the G register.

Subsection 14 (SUB14)

This subsection is executed to cause a parity error for the R-to-Y register transfer.

Subsection 15 (SUB15)

This subsection causes parity errors in the PP memory address at the parity checker.

Subsection 16 (SUB16)

This subsection tests the data to CM with inverted tag-out parity.

Subsection 17 (SUB17)

This subsection tests data-in parity on CM transfers.

Subsection 18 (SUB18)

This subsection tests the parity error reporting when the CM address parity bit is inverted.

Subsection 19 (SUB19)

This subsection tests the disable SECDED code generator to PP memory.

Subsection 20 (SUB20)

This subsection code is use to test for conversion errors.

Subsection 21 (SUB21)

This subsection code is not used.

Subsection 22 (SUB22)

This subsection code is not used.

Subsection 23 (SUB23)

This subsection code is not used.

Subsection 24 (SUB24)

This subsection tests for the CM request/resync error.

Subsection 25 (SUB25)

This subsection tests the force mark line parity bit low.

Subsection 26 (SUB26)

This subsection tests the force tag-out/in parity error.

Subsection 27 (SUB27)

This subsection tests the force response code parity error.

Subsection 28 (SUB28)

This subsection tests the invert function-out parity bit.

Subsection 29 (SUB29)

This subsection tests the force address-out parity bit low.

Subsection 30 (SUB30)

This subsection tests the parity error reporting when data-in parity is forced low on all groups.

Subsection 31 (SUB31)

This subsection tests the force data-out parity bit low.

Subsection 32 (SUB32)

This subsection tests the invert channel parity on channel 17_8, or when the parity is inverted on the channel 15_8 data bus.

Subsection 33 (SUB33)

This subsection tests the invert Maintenance register write parity on the data being written into the register.

Subsection 34 (SUB34)

This subsection tests the invert nanocode parity.

Subsection 35 (SUB35)

This subsection tests the invert read parity bit.

Subsection 36 (SUB36)

This subsection tests parity error reporting when the parity is inverted on the channel 15_8 data bus.

Subsection 37 (SUB37)

This is the test for radial interface read.

Subsection 38 (SUB38)

This is the test for radial interface write.

Subsection 39 (SUB39)

This subsection tests the force lost clock error.

Subsection 40 (SUB40)

This subsection tests the force CM request in the ADU.

Subsection 41 (SUB41)

This subsection tests the force tag-in parity error in the ADU.

Subsection 42 (SUB42)

This subsection tests the force response code parity error in the ADU.

Subsection 43 (SUB43)

This subsection tests the force D5 full in the ADU.

Subsection 44 (SUB44)

This subsection tests the force CMC busy at the CMI.

Subsection 45 (SUB45)

This subsection tests for OS bounds violation when a slave PP tries to write a CM word outside the allowed region specified by OS bounds.

Subsection 46 (SUB46)

This subsection tests the set inhibit PP CM request.

Subsection 47 (SUB47)

This subsection tests the clear inhibit PP CM request.

TPM44 – Two-Port Multiplexer Test

The Two-Port Multiplexer Test (TPM44) is a standalone only test.

TPM44 tests the interface from and to the PP across channel 15_8 and port 1. The test requires user interaction and all faults are detected by the operator when the test stops with a display message error or a hung condition.

The test exercises only port 1 for functions that could interrupt normal PC console operation. The read and write clock sections are run from both ports and therefore test all ports.

The Clock sections test the special features available on the two-port mux.

The following message is displayed only at the beginning of the test when the set PARAM message is displayed:

```
SET DISPLAY TERMINAL

ON LINE,  PAGE MODE,  FULL DUP.

(LOC 141B)=PARITY SELECT

00=ODD,  01=EVEN,  02=NO PARITY

(LOC 142B)=STOP BITS

00=1 STOP BIT,  01=2  STOP BITS.

DEFAULT-EVEN PARITY/2 STOP BITS
```

Section Descriptions

TPM44 is divided into the following ten sections.

Section	Tag	Description
00	SEC00	Tests the Read Deadstart Port and Terminal function.
01	SEC01	Tests the function response and FIFO.
02	SEC02	Tests the DTR and Connect function.
03	SEC03	Tests the loopback with 5-, 6-, 7-, and 8-bit mode.
04	SEC04	Tests the Master-Clear and Disconnect functions.

Section	Tag	Description
05	SEC05	Tests the data in overrun.
06	SEC06	Reads and displays the wall clock.
07	SEC07	Tests the wall clock by writing different test patterns.
08	SEC08	Tests the wall clock increment feature.
09	SEC09	Enters the time in the wall clock.

Section 00 (SEC00)

This section tests the Read Deadstart Port and Terminal function. The deadstart port and terminal are displayed on the console and operator input is required to continue the test.

Subsection	Tag	Description
00	SEC00.0	Reads and displays the deadstart port and terminal type.

Section 01 (SEC01)

The section has the following four subsections.

Subsection	Tag	Description
00	SEC01.4	Checks to see if the test initialization detected a response to the Connect function.
01	SEC01.6	Checks for the response on issued functions.
02	SEC01.8	Checks for FIFO operation.
03	SEC01.9	Checks if the DTR signal remains set if previously set by the DTR function after the FIFO has emptied.

Section 02 (SEC02)

The section has the following four subsections.

Subsection	Tag	Description
00	SEC02.1A	Tests the loopback feature with 8-bit mode.
01	SEC02.1	Tests the loopback feature with 7-bit mode.
02	SEC02.1	Tests the loopback feature with 6-bit mode.
03	SEC02.1	Tests the loopback feature with 5-bit mode.

Section 03 (SEC03)

The section has the following two subsections.

Subsection	Tag	Description
00	SEC03.0	Test the four character input buffer.
01	SEC03.0	Test the data in overrun bit. Verifies that the data in overrun bit will set if the data in the input buffer is overwritten.

Section 04 (SEC04)

The section has the following four subsections.

Subsection	Tag	Description
00	SEC04.2	Tests the clear Data Terminal Ready function (400) and checks for proper status on the port.
01	SEC04.6A	Tests the set Data Terminal Ready function and checks for proper status.
02	SEC04.8	Tests the Connect function (700X) by outputting on one port to fill the buffer and than connecting to the other port without interrupting the original output.
03	SEC04.b	Verifies that the two-port multiplexer will set the Data Terminal Ready status bit when in Output mode.

Section 05 (SEC05)

The section has the following two subsections.

Subsection	Tag	Description
00	SEC05.1	Tests the Master Clear function (700). This subsection will display the message `Enter Character`. The character is to be entered by the operator that is required to fill the input buffer to test the Master Clear function.
01	SEC05.6	Tests the Disconnect function (6000).

Section 06 (SEC06)

The section has the following subsection.

Subsection	Tag	Description
00	SEC060	Reads and displays the calendar clock.

Section 07 (SEC07)

The section has the following subsection.

The subsection tests the Write and Read Wall Clock function by writing a pattern of data to the clock and then reading it back to verify. There are five fields defining the year (YY), month (MM), day (DD), hour (HH), and minute (MN).

```
:---------------------------------:
: 00 : YY : MM : DD : HH : MN : 00 :
:    :    :    :    :    :    :    :
```

Field	Condition	Patterns
*****	*********	********
YY	0	00
YY	1	01
.	.	.
.	.	.
YY	98	98
YY	99	99
MM	100	01
MM	101	02
.	.	.
.	.	.
MM	110	11
MM	111	12
DD	112	01
DD	113	02
.	.	.
.	.	.
DD	141	30
DD	142	31
HH	143	00
HH	144	01
.	.	.
.	.	.
HH	165	22
HH	166	23
MN	167	00
MN	168	01
.	.	.
.	.	.
MN	225	58
MN	226	59

In each condition, a pattern is written in a field; other fields contain 01. For example:

Condition 03: 00 03 01 01 01 01 00
Condition 98: 00 98 01 01 01 01 00
Condition 108: 00 01 09 01 01 01 00.

Section 08 (SEC08)

The section has the following three subsections.

Subsection	Tag	Description
00	SUBZER	This subsection checks propagation delay from minute-to-hour to day-to-month and to-year.

NOTE

To run subsections 1 and 2, clear quick look mode bit.

01	SUBONE	This subsection checks the propagation of the carry (the increment of the clock). minute → hour, hour → day, day → month, month → year.
02	SUBTWO	This subsection tests the calendar clock to make sure that the last day is right for each month.

Section 09 (SEC09)

The section has the following subsection.

Subsection	Tag	Description
00	SEC090	This subsection can be used to set date and time in the wall clock from the keyboard. When the message ENTER DATE/TIME YYMMDDHHMN is displayed, enter data through the keyboard in the above form, where YY = year, MM = month, DD = day, HH = hour, MN = min. No separators are required. For example: 8506100812. If data is illegal an INVALID ENTRY message is displayed. If good data is entered, it is written to the clock, read back and displayed. Type ABS **CR** to exit or abort this subsection at the end of the test.

Isolation Diagnostics

The isolation diagnostic (FII44) analyzes errors reported by the detection tests and attempts to identify a group of logic paks that are likely to be causing the problem.

FII44 - Fault Isolation Program

The Fault Isolation Program (FII44) is a table driven analyzer that uses data recorded by the IOUCP Error Processing Routine (ERRCP) during IOU test execution.

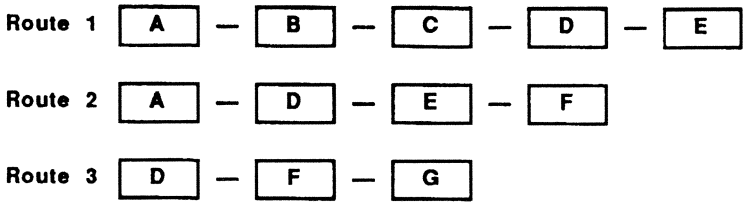
ERRCP records the following three classes of errors and assigns route numbers to them.

- Unexpected parity errors - (route 0)
- Invert parity errors - (1 through 77_8)
- Data/control errors - (100 through 177_8)

FII44 analyzes the errors recorded by the detection tests using the route numbers as the base for isolation.

The route numbers for fault isolation are an index to a predefined string of modules.

Example:



FII44 sets error location (ERROR) to zero and starts by processing route number 0. If an error is found in this route number, it sets error location (ERROR) to nonzero. If no errors are found for route number 0, FII44 processes route numbers 100 through 177_8. If no errors are found for route numbers 100 through 177_8, FII44 processes errors for route numbers 1 through 77_8.

When FII44 finds errors in the route numbers, it calls subroutines to find and identify the pak associated with the highest number of error routes. The subroutine assigns priority number one to paks with the highest count and then identifies lower priority paks. The maximum number of paks that can be called is 10.

Log Errors Routine

This routine is built into the IOUCP4. It is used to log errors in the error buffer for fault isolation. It is called by ERRCP. The following classes of errors are logged.

- Unexpected parity error: When a parity error is detected, IOUCP4 generates a ROUTE=0 code and passes the contents of Fault Status registers 1 and 2 (FS1 and FS2) in (PEBUF) to the Error Log Routine.

- Invert parity error: When MRT44 detects an error, it generates a ROUTE=1 through 77_8 code and passes the logical difference between expected and received FS1 and FS2 in PEBUF to the Error Log Routine. The following parameters are also passed to the Error Log Routine.

  (EXPAT) Expected Pattern Buffer
  (RCPAT) Received Pattern Buffer
  (PPNUM) Failing PP Number
  (CHNUM) Failing CH Number
  (PEBUF) Parity Error Buffer

- Data/control error: When tests detect an error, they generate a ROUTE=100 through 177_8 code and passes the following parameters to the Error Log Routine:

  (EXPAT) Expected Pattern Buffer
  (RCPAT) Received Pattern Buffer
  (PPNUM) Failing PP Number
  (CHNUM) Failing CH Number
  (PEBUF) Parity Error Buffer

Figure 2-1 shows the Monitor PP map. The errors are recorded by Error Log Routine in the Error Log Buffer and processed by FII44.

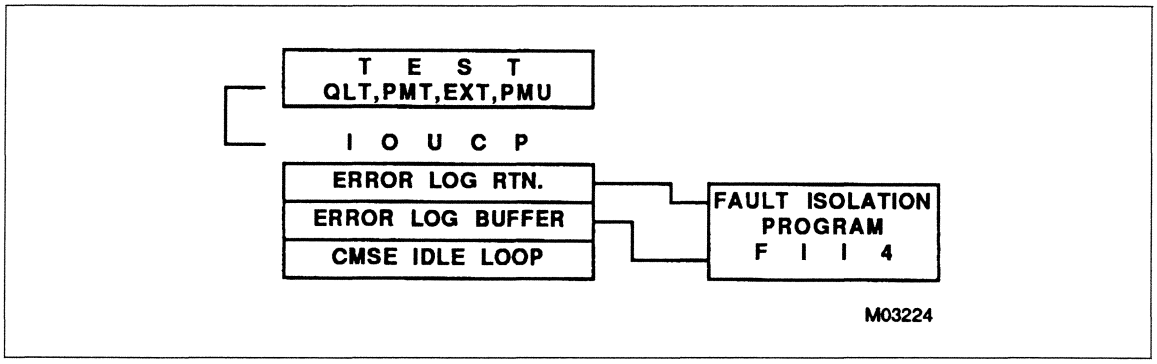


Figure 2-1. Monitor PP Map

Unexpected Parity Error Recording

The route number=0 is transmitted whenever unexpected parity errors occur. As shown in figure 2-2, the errors are logged by setting the RPEBUF equal to the accumulative logical difference of FS1 and FS2 from PEBUF, and PCOUNTER is incremented by one.

The size of the Recorded Parity Error Buffer (RPEBUF) is 20_8 locations.

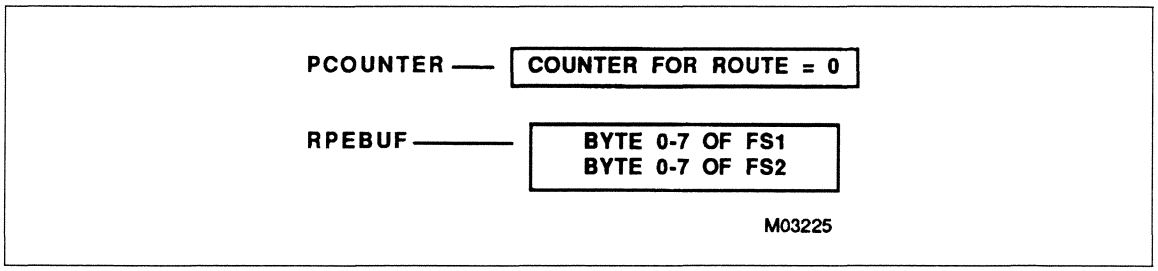


Figure 2-2. Parity Error Buffer

Invert Parity Error Recording

The routes numbered 1 through 77_8 are the invert parity error routes. When one of these errors is reported, the corresponding location (PCOUNTER+(ROUTE)) is incremented by one.

Also, a cumulative logical difference of FS1 and FS2 is recorded in location RPEBUF+20_8 to location RPEBUF+37_8.

Figure 2-3 shows what these locations represent.

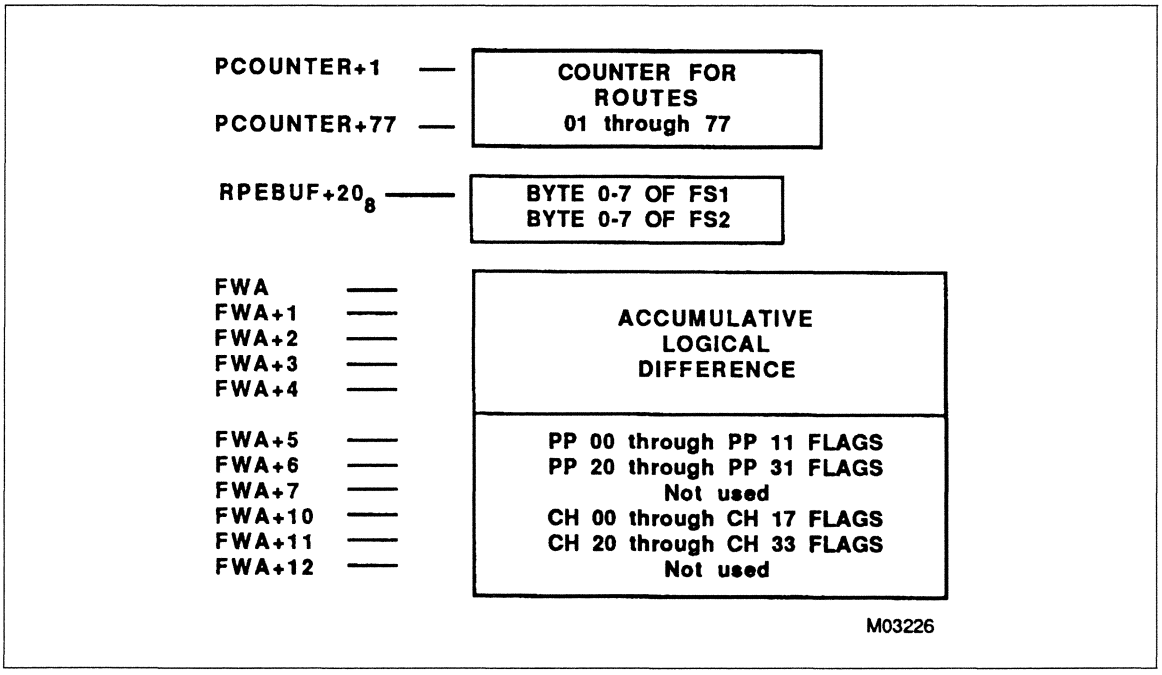


Figure 2-3. Invert Parity Error Buffer

Control and Data Error Recording

The routes numbered 100 through 177_8 represent the errors for data and control flow. The route numbers are used to generate the starting address for cumulative logical difference, PP, and channel flags when an error is detected.

The logical difference is calculated using expected (EXPAT) and received (RECPAT) patterns. The PP and channel position flags are calculated using (PPNUM) and (CHNUM).

The corresponding counter for failing route (PCOUNTER+(ROUTE)) is incremented by one.

Figure 2-4 shows the buffer for one failing route number.

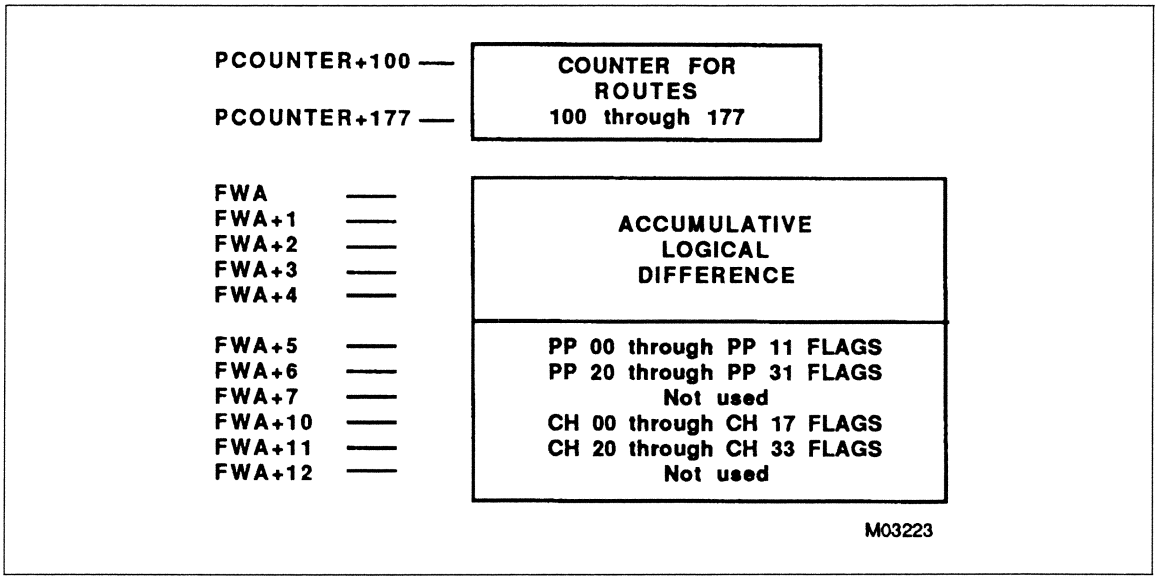


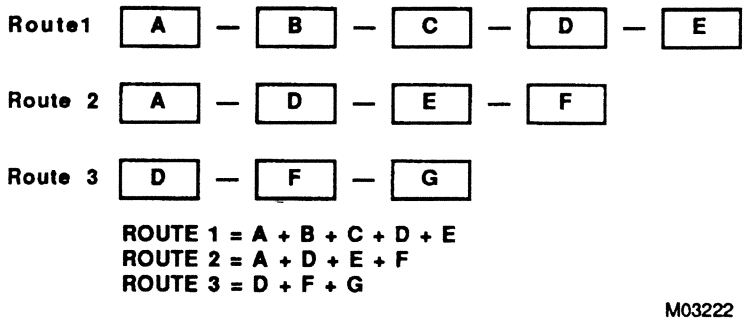
Figure 2-4. Control and Data Error

Fault Isolation Program

FII44 analyzes the errors recorded by the detection tests using route numbers as a base for isolation.

The route numbers for FII44 are considered as a index for the string of the involved modules.

Example:



FII44 sets location (ERROR) to zero, which is used as a flag when an error condition is detected.

The routine starts with the processing of the route numbers 100 through 177_8. These route numbers represent the data and control flow errors. This processing is done by checking the corresponding location in the counter buffers for nonzero.

When a nonzero is found, location (ERROR) is set to nonzero.

If no error for these routes was found, it processes the unexpected parity error route (Route=0). If the counter for this route is 0, it processes the invert parity routes (1 through 77_8). If there are no errors for these routes, it sends the message to CMSE: NO ERROR RECORDED.

If location (ERROR) is nonzero, Fault Isolation Control Program (FICP) calls FISORT routine to find the pak with the highest number of calls. The counter for that pak has the highest count.

The FISOL routine is then called to set priority and identify the pak type and location. The calling table is passed to CMSE to display the message SUSPECTED FAILING MODULE/S (FISM1, Fault Isolation Message 1).

Data and Control Processing

The errors with route numbers 100 through 177_8 are processed in the following manner:

- Use COPYER routine to set the following:
  - Accumulative logical difference to received buffer.
  - Failing PP flags in PP parameter area.
  - Failing channel flags in channel parameter area.
  - Number of failing barrels (NUMBAR).
  - Failing physical barrel number (BARNUM).
- Call the processor for that route number to:
  - Determine if it is a data or control failure.
  - Find the number of failing bits and failing bit numbers.
  - Find the failing channel number and number of failing channels if it is channel failure.
  - Increment counters for the paks that caused the error.

Invert Parity Processing

This routine checks if any location in the PCOUNTER buffer is nonzero and, if not, returns control to the Fault Isolation Control Program (FICP) with location (ERROR) unchanged. It sets location (ERROR) to nonzero if any failing route is found. If there is any error, it calls COPYPE routine to copy the recorded accumulative logical difference to PEBUF and:

- Sets NUMBAR.
- Sets BARNUM.

The routine then increments the counters for involved paks by one.

Parity Error Processing

This routine checks if location PCOUNTER is nonzero and, if not, returns control to FICP with location (ERROR) unchanged. If PCOUNTER is nonzero, error location (ERROR) is set to nonzero and the COPYPE routine is called to copy the recorded accumulative logical difference for this route in PEBUF and:

- Sets NUMBAR.
- Sets BARNUM.

The parity error processor then increments the involved paks by one.

Module Sorting

The Pak Sorting Routine (FISORT) is used to find the pak counter with the highest count and set Maximum Count Value (MAXCNT).

If the highest count is zero (processors were not able to determine the failing pak multiple uncorrelated error), the FISM2 (Fault Isolation Message 2) is passed to CMSE.

Fault Isolation

The Fault Isolation Routine (FISOL) is called to identify the pak type and pak location and to set the priority for the paks that have counters nonzero.

FISOL assigns priority number one to all the paks with the highest counters. It then decrements the maximum count and increments the priority number for the next set of paks with the same count as the current maximum count.

FISOL calls MODID for the pak identification and location.

The maximum number of paks to be called is 10.

LDT – Long Deadstart Sequence

The Long Deadstart Sequence (LDT) is a quick look program that resides in ROMs attached to upper core of logical PP00 memory (5400 through 7777_8). LDT transfers to barrel 1 after barrel 0 completes. LDT continues to transfer to barrels 2 and 3 if they are installed. PPs within each barrel may be reconfigured by using the deadstart display and entering the RP x command. Initiating the long deadstart sequence sets the following conditions:

- A master clear is issued to all PPs and all error status bits are cleared.
- PP00 is forced to read the test program stored in ROM.
- PP00 address is forced to 6000_8 (5400 through 5700_8 is for test update and transfer routine).
- A PP register display is brought up on the right-hand side of the screen.

While LDT is executing, certain error status bits are monitored to determine if the hardware has been detected. If a hardware failure is detected, a message is displayed below the PP register.

Each error stop is indicated by a unique P register address. The customer engineer determines what error condition was detected by examining the P register display and referring to the LDT error tables. Refer to the CYBER 960 Series and 970 Series System Troubleshooting Guide listed in the preface of this manual.

When execution of LDT is finished (all installed barrels run), the short deadstart sequence is initiated using the program currently displayed on the screen.

LDT Tests

LDT performs the following sequence of tests:

1. Arithmetic Unit Test
2. PP00 Memory Test (up to location 5377_8)
3. One Word Channel Test
4. Block I/O Transfer Test
5. Block I/O Transfer with Conversion Test
6. Multi-PP Arithmetic Unit Test
7. LDT Bit Test
8. Detection Tests

The following paragraphs describe these tests.

Arithmetic Unit Test

The tests performed by the arithmetic unit test are:

Test	Description
Unconditional Jump Test	Checks execution of the 03 instruction and the lower 6 bits of the d portion of the instruction. Error stops are: – PP stops on 0300 or 0377 instruction for single Q-bit failure. – Hardware timeout counter error indicates that a multiple Q-bit failure occurred or P register failed.
Conditional Jump Test	Checks instructions 04 through 07 with A equal to 0, A equal to 1, and A equal to minus 0. Error stops are on 03 through 07 instructions.
1x Instructions Test	Checks instructions 10 through 17 using the lower 6 bits of the A register. Error stops are on branch instructions.
2x Instructions Test	Checks instructions 20 through 25, arithmetic operations in the A register with all 18 bits, and shift network. Error stops are on branch instructions.
3x Instructions Test	Checks instructions 30 through 37 using the lower addresses of PP00 memory for store instructions. Error stops are on branch instructions.
4x Instructions Test	Checks instructions 40 through 47 and all 16 bits of the Q register. Lower addresses of PP00 memory are used to check restore capability. Error stops are on branch instructions.
5x Instructions Test	Checks instructions 50 through 57 with and without the Q flag set. Error stops are on branch instructions.
13x Instructions Test	Checks execution of instructions 130 through 137 with 16-bit operands. Error stops on branch instructions.
14x Instructions Test	Checks execution of instructions 140 through 147 with 16-bit operands. Error stops are on branch instructions.
15x Instructions Test	Checks execution of instructions 150 through 157 with 16-bit operands, with and without the Q flag set. Error stops are on branch instructions.
RJM and LJM Instructions	Checks execution of instructions. Test 01 and 02 using previously written addresses in lower core of PP00 memory. Error stops are on branch instructions.

PP00 Memory Test

This test executes from low addresses of PP memory. The test is transferred from ROMs to PP memory and checked for correct transfer. Error stop is on a branch instruction (PP00 at location 24_8, PP01 through PP04 at PP00 memory location 210_8 plus PP number).

The test then checks PP memory with 10 fixed data patterns and addresses. Error stop is on a branch instruction. The contents of the A register represent the failing bits. The failing pattern and failing address are stored in low addresses of PP00 memory.

PPM00 Location	Contents
34_8	Failing bits
35_8	Failing patterns
36_8	Failing addresses

One Word Channel Test

This test checks channels with one word input/output using all possible bit combinations and channel status during input/output.

Barrel Number	Channel Tested (Octal)
0	0 through 4, 13, 15, 17
1	5 through 11, 12, 15, 17
2	12, 15, 17, 20 through 24
3	12, 15, 17

Barrel 3 does not test CIO channels. Error stops are on branch instructions and are unique for each channel status and data error.

Block I/O Transfer Test

This test checks inter-PP communication channel 12 using a slightly modified PP memory test. A timeout counter protects PP00 from a hung condition. Error stops are on branch instructions that are unique for each channel/PP.

Block I/O Transfer with Conversion Test

This test checks inter-PP communication with 12-to-16-bit and 16-to-12-bit conversion over channel 12. Error stops are on branch instructions. A timeout feature protects PP00 from a hung condition.

Multi-PP Arithmetic Unit Test

This test is the same as the Arithmetic Unit Test described previously, but runs in PP01 through PP04 simultaneously. If an error is detected, the PP in which the test is running stops and P register stop lights are the same as for the Arithmetic Unit Test. PP00 is in a loop waiting for a message from PPxx indicating that the test successfully passed. The looping address is unique to each PP. Each PP is force deadstarted, waits for the channel 13 flag to set before starting, and sets the channel 12 flag when completed.

LDT Bit Test

This test checks the LDT bit in the maintenance register. If the bit is clear, PP00 stops and the P register indicates an error in the maintenance register.

Detection Tests

The detection tests are a series of programs that test each PP and channel in the IOU4S for correct functional operation. These tests also record errors such as differences between expected and actual (received) results and generate internal codes that identify the area of the IOU4S under test when the error was detected.

QLT42 – Quick Look Test

The Quick Look Test tests one-word input/output over each channel. The test uses hardware features such as force idle PP and force deadstart PP.

At the end of the test, all tested channels are set active, empty, and all associated PPs, if selected, are in deadstart condition.

Section Descriptions

The Quick Look Test section numbers, converted to octal, represent the channel under test (that is, section 8 represents channel 10_8). For a channel to be tested, both the section and the channel must be selected by bits in the section select and channel flag parameter words. Refer to parameters and control words section of the MSL 153/155 IOU Test Procedures Reference manual listed in the preface of this manual.

This section has the following four subsections:

Subsection	Tag	Description
00	SUBSEC00	Initializes the associated PP (force idle PP).
01	SUBSEC01	Tests one word I/O over tested channel.
02	SUBSEC02	Tests channel status test with escape bit (Q bit 2^5) set.
03	SUBSEC03	Forces PP in deadstart condition.

At the end of the section, channel status is active, empty, and the associated PP is in deadstart condition.

Subsection 00 (SUBSEC00)

Subsection 00 has two conditions that are executed only if the associated PP for the channel has been selected.

Condition	Description
00	Forces PP idle through the Maintenance register, reads K register of the associated PP, and checks for 1077_8 indicating that the PP is idle.
01	Tests channel flag operation.

Subsection 01 (SUBSEC01)

Subsection 01 tests one word I/O over each channel selected to be tested and checks the status of the channel during the I/O transfer. This subsection has 1000_8 conditions, that use different patterns and complements of that pattern. There are 152_8 fixed patterns and 626_8 randomly generated patterns.

Each condition uses one pattern and the complement of the pattern. During this I/O, the test checks for correct channel status and channel error flag clear status.

Subsection 02 (SUBSEC02)

Subsection 02 has the following nine conditions.

Condition	Description
00	Disconnects already inactive channel with escape bit set.
01	Outputs one word on inactive channel with escape bit set.
02	Inputs one word from inactive channel with escape bit set.
03	Blocks input over inactive channel expecting that the first word address of the block input is cleared.
04	Blocks output over inactive channel expecting that the contents of A are not changed.
05	Activates already active channel with escape bit set.
06	Functions FAN on active channel with escape bit set.
07	Functions FNC on active channel with escape bit set.
08	Forces channel active and full and checks for empty when channel is deactivated.

NOTE

It is possible for subsection 02 to cause the PP from which the test is being executed to hang if the escape feature (bit 2^5 of I/O instructions) does not work. There is no way to prevent this hang if the feature fails.

Subsection 03 (SUBSEC03)

This subsection has two conditions. Condition 01 is executed only if the channel has an associated PP and PP is selected.

Condition	Description
00	Tests channel for active and empty status.
01	Forces PP to deadstart condition. Check P register of the associated PP for 177777_8 value. Check K register of the associated PP for 71_8 value. Check Q register of the associated PP for channel number. Check A register of the associated PP for 20000_8 value.

PMT42 - PP Memory Test 1

PP Memory Test 1 (PMT42) checks the operation of the PP memories in all barrels not used by the IOU Control Program (IOUCP2). The test does not rely on code executing in the PP memory being tested. PP memories are tested sequentially using force deadstart, idle, and force dump. Since the barrel being used by IOUCP2 was checked by the LDT tests, PMT42 assumes that it is operational.

Section Descriptions

The PP Memory Test 1 section numbers, in decimal, represent the PP memory being tested. For a memory to be tested, both the section and the PP must be selected by bits in the section select and PP flag parameter word. Refer to the parameters and control words section of the MSL 153/155 IOU Test Procedures Reference manual listed in the preface of this manual.

All data transfers for PPs are done over the communication channel (channel 12_8 or 13_8).

This section has 32 subsections numbered 00 through 31, as described in the following table.

Subsection	Tag	Description
00	B00.00	Forces deadstart PP with check.
01	B01.00	Checks P and A registers.
02 to 30	B02.00	Tests memory with patterns.
31	B03.00	Tests memory with address patterns.

At the beginning, all PPs are forced into a hung condition by a force idle feature. The PPs are left idle at the end of each section, but are forced to a deadstart condition at the end of the test.

Subsection 00 (B00.00)

This subsection has the following five conditions:

Condition	Description
00	Forces PP to deadstart condition.
01	Checks P register of tested PP for the value 177777_8.
02	Checks Q register of tested PP for deadstart channel value.
03	Checks K register of tested PP for the value 007100_8.
04	Checks A register of tested PP for the value 020000_8.

Subsection 01 (B01.00)

This subsection has the following condition:

Condition	Description
00	Repeats one word output to fill tested memory. Checks the P and A registers of the PP being tested for correct count after each word of output.

Subsections 02 through 15 (B02.00)

These subsections output standard test patterns to complement every second word, then dump the test PP over the used channel to the monitor PP to check the data.

These subsections have the following two conditions:

Condition	Description
00	Outputs data blocks of 100_8 words to fill up the test PP memory.
01	Forces a dump of the test PP and input one word at a time checking data.

Subsections 16 through 30 (B02.00)

These subsections use the same code as subsections 01 through 15, but use random number patterns.

Subsection 31 (B03.00)

This subsection outputs the address of the corresponding memory location to fill the test PP memory, then dumps the test PP over the used channel to the monitor PP to check the data.

This subsection has the following two conditions:

Condition	Description
00	Repeats one word output to fill the memory of the test PP.
01	Forces a dump of the test PP and inputs one word at a time checking data.

EXT42 – Execution Unit Test

The Execution Unit Test checks instruction execution and the arithmetic units in PPs other than the one in which the IOUCP2 and CMSE are running. CMSE loads EXT42 into the same PP with IOUCP2, inserts the communication channel in the communication routine, and distributes the updated copies to the PPs to be tested. The test is started by deactivating the communication channel for the assigned PP.

Section Descriptions

EXT42 is divided into the following three sections:

Section	Tag	Description
00	UJNTST	Tests arithmetic unit and execution of instructions 00 through 57_8 (12-bit operand instructions).
01	TEST13X	Tests arithmetic unit and execution of instructions 1030 through 1057_8 (16-bit operand instructions).
02	STSTEST	Tests execution of the channel instructions.

Section 00 (UJNTST)

This section tests 12-bit operand instructions 00 through 57_8 and is divided into the following six subsections.

Subsection	Tag	Description
00	UJNTST	Tests execution of 00 through 07 instructions.
01	TEST1X0	Tests execution of 10 through 17 instructions, and A register bits 2^0 through 2^5.
02	TEST2X0	Tests execution of 20 through 23 instructions and A register bits 2^0 through 2^{17}.
03	TEST3X	Tests execution of 30 through 37 instructions and restore ability.
04	TEST4X	Tests execution of 40 through 47 instructions and A register bits 2^0 through 2^{11}.
05	TEST5X	Tests execution of 50 through 57 instructions and Q adder.

Section 01 (TEST13X)

This section tests 16-bit operand instructions and is divided into the following four subsections.

Subsection	Tag	Description
00	TEST13X	Tests execution of 1030 through 1037 instructions and restore ability on 16-bit boundary.
01	TEST14X	Tests execution of 1040 through 1047 instructions and restore ability using indirect addressing.
02	TEST15X	Tests execution of 1050 through 1057 instructions using index addressing.
03	JMPTEST	Tests execution of 01 and 02 instructions.

Section 02 (STSTEST)

This section tests I/O instructions (channel instructions) and has the following four subsections.

Subsection	Tag	Description
00	STSTEST	Tests 64 through 67 instructions.
01	IANTEST	Tests 70, 72, 74, and 75 instructions.
02	IOTEST	Tests 1064 through 1067 instructions.
03	ESCTEST	Tests channel instructions with escape bit (Q bit 2^5) set.

PMU42 - PP Memory Test 2

PP Memory Test 2 (PMU42) performs a more rigorous check of the PP memories than PP Memory Test 1 (PMT42). The advantage over PMT42 is that PMU42 sends copies of the test code to the PP memories to be checked. This provides more variety in the way memory is referenced and allows more strenuous code to be used.

PMU42 allows the parameters to be changed for each copy to be loaded to a test PP, provided parameter change capability is chosen initially.

Section Descriptions

The section, subsection, and condition structure of PMU42 is defined in reference to where the code executes in a particular test PP. The code in the test PP is executable from either the lower or upper half of memory, so that the entire memory can be tested.

Selection of sections 00 and 01 tests the entire memory.

Section	Description
00	Executes in the lower half of memory and tests all memory except test core and direct cells used by the lower half copy.
01	Executes in the upper half of memory and tests direct cells and core used by the lower half copy.

Each section includes all of the following 48 subsections except as noted.

Subsection	Tag	Description
00	SS00	Checks addressing to find a solid addressing failure using a quick look check. This is not affected by the memory bounds of a section.
01 through 36	SS01 through 36	Checks available test memory using standard and random patterns.
37 through 39	SS37 through 39	Tests available test memory with a random word in each location.
40 through 46	SS40 through 46	Loads random addresses rapidly in a prearranged sequence to check addressing.
47	SS47	Checks if faults are induced when address lines are selectively toggled. This is not affected by memory bounds of a section.

Subsection 00

This subsection uses the following two conditions to check for solid address failures:

Condition	Description
00	Initializes location 0 with all zeros, and stores ones in all remaining locations with a one-bit-set address. A nonzero word in location 0 indicates an address line shorted.
01	Initializes location 7777_8 with all ones, and stores zeros in all remaining locations with a one-bit-clear address. A zero-bit in location 7777_8 indicates an open address line.

Subsections 01 through 06

These subsections use a standard fixed pattern. The testable memory is checked in groups of five words with the pattern, then its complement, being loaded into alternate locations. The words are quickly read back and checked in sequential order.

Subsections 07 through 14

These subsections are the same as subsections 01 through 06, except the standard patterns are circularly shifted. These sliding patterns use 16 conditions, one for each circular shift.

Subsections 15 through 21

These subsections are exactly the same as subsections 01 through 06. The placement allows staggering of fixed and sliding pattern checks.

Subsections 22 through 36

These subsections are the same as subsections 01 through 06, except the patterns are randomly chosen before the tests are copied to the test memories.

Subsections 37 through 46

These subsections have one condition. Each subsection uses a real-time-clock input to generate a series of 1000_8 addresses. For each address the previous address is used as data. This daisy chain is then rapidly retraced using load instructions. If no error is generated by rapid changes of the address/data lines, the 1000th load is the reference point.

Repeating the condition uses the same seed for the random number generation; repeating the subsection uses a new seed for each cycle.

The subsections are duplicated to allow multiple cycles without selecting a repeat subsection.

Subsection 47

This subsection checks if the address lines are affected by crosstalk. An instruction sequence is executed that causes the address lines to switch from all zeros to all ones, except for one bit to be checked for staying zero. This is repeated for all 13 address bits then another sequence is used for checking all ones toggling to all zeros (but one).

This subsection uses the following two conditions:

Condition	Description
00	Toggles all address lines, except the test bit, from zeros to ones by executing a store with indirect addressing through location 0 that contains a one-bit-clear address.
01	Toggles all address lines, except the test bit, from ones to zeros. This is accomplished by executing an indexed long jump instruction, from location 7776_8 to location 0 plus a one-bit-set address.

CHD42 - Channel Test

The Channel Test (CHD42) checks inter-PP data transfer over each tested channel.

The test uses a pair of PPs for inter-PP transfer. One PP is called the PP transmitter and the other, the PP receiver. The PP transmitter communicates through the Double PP Driver (DPPD) with the IOUCP2 over the communicating channel. The PP receiver communicates with the PP transmitter over the testing channel.

The PP transmitter and the PP receiver are randomly selected at the beginning of the test and whenever the test is repeated.

The execution of the complete test with one channel is considered a section. The number of selected sections specifies the number of the channels to be tested for one PP pair (default is 32 sections selected).

When all selected sections are done, the end of the test is reported and, if repeat test is selected new pairs are randomly generated.

The maximum number of words (size of the I/O block) to be transferred is 1000_8.

Section Descriptions

In this test, a section is the execution of the complete program using one pair of PPs that are testing one channel. The PP receiver returns received data over the channel being tested to the PP transmitter. The PP-transmitter section is divided into the following five subsections.

Subsection	Tag	Description
00	SUBSEC00	Tests one word I/O over tested channel using OAN instruction.
01	SUBSEC01	Tests one word I/O over tested channel using OAM instruction.
02	SUBSEC02	Tests the channel flag operation (PP-receiver number is the channel number for channel flag test).
03	SUBSEC03	Tests block I/O over tested channel using a OAM instruction.
04	SUBSEC04	Tests block I/O over tested channel using a OAPM instruction (conversion).

Subsection 00 (SUBSEC00)

Subsection 00 tests the data and channel status during one word I/O using a single word output (OAN) instruction over the channel being tested. The data used for I/O are word counts. Every condition is represented by one word count.

Subsection 01 (SUBSEC01)

Subsection 01 tests the data and channel status during one word I/O using an output block (OAM) instruction over the channel being tested. The data used for I/O are data patterns used in subsections 03 and 04. Every condition represents one pattern.

Subsection 02 (SUBSEC02)

Subsection 02 cannot be deselected. If subsections 03 or 04 are selected, then subsection 02 is selected. It tests the operation of the channel flag (channel flag number is equal to the receiving PP number). This channel flag is used in subsections 03 and 04 to control the receiving PP.

Subsection 03 (SUBSEC03)

Subsection 03 tests the data and channel status during block I/O using a output block (OAM) instruction over the channel being tested. The size of the block is represented by the word count. Every condition is one block size (one word count).

Subsection 04 (SUBSEC04)

Subsection 04 tests the data and channel status during block I/O using output block (OAPM) instruction with conversion over the channel being tested. The size of the block is represented by the word count. Every condition is one block size (one word count).

CMA42 - Central Memory Access Test

The Central Memory Access Test (CMA42) checks the data path from PP memory to central memory (CM) and from CM to PP memory. It also checks the Relocation (R) register which, in conjunction with the A register, makes the CM address.

Section Descriptions

CMA42 is divided into ten sections:

Section	Tag	Description
00	SEC00	Tests R register.
01	SEC01	Tests (R+A), CM address.
02	CWDTST	Tests read/write CM with 60-bit single CM word.
03	ADRTST	Tests read/write addresses in CM with 60-bit single CM word.
04	CWMTST	Tests read/write CM with 64-bit single CM word.
05	CLDTST	Tests read/write block in CM with 60-bit CM word.
06	CLMTST	Tests read/write block in CM with 64-bit CM word.
07	RSLTST	Tests read and set lock.
08	RCLTST	Tests read and clear lock.
09	MIXTST	Tests read/write in mixed mode (60- and 64-bit CM word, single CM word and block).

Section 00 (SEC00)

This section has the following three subsections:

Subsection	Tag	Description
00	RRG00	Tests the lower 12 bits of R register, with the upper 10 bits of R set to zero.
01	RRG01	Tests the upper 10 bits of R register, with the lower 12 bits of R set to zero.
02	RRG02	Tests full R register.

Every subsection has 1000_8 conditions and every condition is a different pattern.

Section 01 (SEC01)

This section has the following subsection:

Subsection	Tag	Description
00	RAATST	Tests full A+R register.

This subsection has 1000_8 conditions and every condition is a different pattern.

Sections 02 through 09

Sections 02 through 09 have 16 subsections each. Subsections 00 through 07 use different starting addresses in CM and do not use the R register. Subsections 08 through 15 use randomly selected R-register values.

Every subsection has 1000_8 conditions and every condition is a different pattern with a different starting address in CM.

For the block read/write, every condition is a different CM word count.

MRA42 – Maintenance Register Access Test

The Maintenance Register Access Test (MRA42) checks Maintenance register (MR) access from all PPs. It also checks the Maintenance Channel (MCH) hardware interlock and MCH priority circuits.

Section Descriptions

MRA4 has the following 26 sections.

Section	Tag	Description
00	SEC00	Tests the MCH interlock and PP priority.
01 through 25		Tests the MCH access from all PPs.

Section 00 (SEC00)

This section has the following two subsections.

Subsection	Tag	Description
00	S00B00	Tests that a PP from the lower barrel has priority over other PPs.
01	S00B01	Tests that only one PP can access the MCH at a time, using MCH interlock.

Sections 01 through 25

Each section tests a MR access from the corresponding PP. Each section has the following five subsections.

Subsection	Tag	Description
00	FUNRES	Tests the response on function. Each condition is one function code.
01	ECHO	Tests the data to and from MR using an echo function. Each condition is one pattern.
02	WRITE	Tests the Write function. The testing PP writes MR and the monitor PP reads and compares values.
03	READ	Tests the Read function. Each condition is one register in MR.
04	SSUMF	Tests the Status-Summary function by reading data using the Status-Summary function and normal read operation.

MRT42 – Maintenance Register Test

The Maintenance Register Test (MRT42) checks all parity networks in the IOU using invert parity feature. All the force conditions that induce errors to other parts of the hardware are also checked.

The 16-digit expected and received messages are the expected and received contents of Fault Status 1 and Fault Status 2 registers.

Section Descriptions

Each PP is a section in test MRT42.

Subsection Descriptions

MRT42 is divided into the following 48 subsections:

Subsection	Tag	Description
00	SUB00	Tests for no reported errors.
01	SUB01	Tests invert parity on channel data to PPU.
02	SUB02	Tests invert parity from PPU to channel.
03	SUB03	Tests invert PPM to R parity.
04	SUB04	Tests invert parity at PPM data checker.
05	SUB05	Tests invert microcode parity.
06	SUB06	Tests invert PPM parity at parity generator.
07	SUB07	Tests invert parity on CM function code.
08	SUB08	Tests invert parity on Y register at parity generator.
09	SUB09	Tests invert parity on A register at parity generator.
10	SUB10	Tests invert parity on shift ROM at parity checker.
11	SUB11	Tests invert parity on Q register at parity generator.
12	SUB12	Tests invert parity on P register at parity generator.
13	SUB13	Tests invert parity on G register at parity generator.
14	SUB14	Tests invert R register to Y parity at checker.

Subsection	Tag	Description
15	SUB15	Tests invert PPM address parity at checker.
16	SUB16	Tests invert tag-out parity at generator.
17	SUB17	Tests invert data-in parity at generator.
18	SUB18	Tests invert CM address parity.
19	SUB19	Tests disable single-error-correction/ double-error-detection (SECDED) code to PP memory.
20	SUB20	Tests invert channel input parity at checker.
21	SUB21	Tests for no error reported.
22	SUB22	Tests for no error reported.
23	SUB23	Tests for no error reported.
24	SUB24	Tests force CM request/resync error.
25	SUB25	Tests force mark line parity bit to zero.
26	SUB26	Tests force tag-out/in parity error.
27	SUB27	Tests force response code parity error.
28	SUB28	Tests invert function-out parity bit.
29	SUB29	Tests force address-out parity bit low.
30	SUB30	Tests force data-in parity bit low.
31	SUB31	Tests force data-out parity bit low.
32	SUB32	Tests invert channel parity from the two-port mux or the maintenance channel.
33	SUB33	Tests invert Maintenance register write parity.
34	SUB34	Tests invert nanocode parity.
35	SUB35	Tests invert read parity bit.
36	SUB36	Tests invert channel 15 data-bus parity at checker.
37	SUB37	Tests invert R/I read-data parity.
38	SUB38	Tests invert R/I write-data parity.
39	SUB39	Tests force lost clock error.
40	SUB40	Tests force CM request in the ADU.
41	SUB41	Tests force tag-in parity error in the ADU.

Subsection	Tag	Description
42	SUB42	Tests force response code parity error in the ADU.
43	SUB43	Tests block d5 full in the ADU.
44	SUB44	Tests force CMC busy at CMI.
45	SUB45	Tests invert OS bounds address parity at checker and the operation of the OS bounds register.
46	SUB46	Tests set inhibit PPU to CM request.
47	SUB47	Tests clear inhibit PPU to CM request.

Subsection 00 (SUB00)

This subsection tests that no errors are reported when a code of (00) is forced in the Test Mode register.

Subsection 01 (SUB01)

This subsection tests for invert parity on a channel to PPU data transfer.

Subsection 02 (SUB02)

This subsection tests for invert parity on a PPU to channel data transfer.

Subsection 03 (SUB03)

This subsection is executed to cause parity error for PPU to R register.

Subsection 04 (SUB04)

This subsection is executed to cause parity error at the PPU memory data checker.

Subsection 05 (SUB05)

This subsection causes parity errors in microcode and the G register.

Subsection 06 (SUB06)

This subsection tests the parity network for data sent to the PP memory.

Subsection 07 (SUB07)

This subsection tests the parity error on CM function.

Subsection 08 (SUB08)

This subsection tests the parity network of the Y register.

Subsection 09 (SUB09)

This subsection tests the A register parity error network.

Subsection 10 (SUB10)

This subsection shifts the data in A to cause shift control ROM parity error.

Subsection 11 (SUB11)

This subsection shifts the data for checking parity in the Q register.

Subsection 12 (SUB12)

This subsection cause parity errors in the P register.

Subsection 13 (SUB13)

This subsection causes a parity error in microcode and the G register.

Subsection 14 (SUB14)

This subsection is executed to cause parity error for R-to-Y transfer.

Subsection 15 (SUB15)

This subsection causes parity errors in the PPU memory address at the parity checker.

Subsection 16 (SUB16)

This subsection tests the data to CM with inverted tag-out parity.

Subsection 17 (SUB17)

This subsection tests data-in parity on CM transfers.

Subsection 18 (SUB18)

This subsection tests the parity error reporting when CM address parity bit is inverted.

Subsection 19 (SUB19)

This subsection tests the disable SECDED code generator to PP memory.

Subsection 20 (SUB20)

This subsection tests the invert parity of the channel input data at the input to the 12/16 conversion array.

Subsection 21 (SUB21)

This subsection tests that no errors are reported when a code of 25_8 is found in the Test Mode register.

Subsection 22 (SUB22)

This subsection tests that no errors are reported when a code of 26_8 is found in the Test Mode register.

Subsection 23 (SUB23)

This subsection tests that no errors are reported when a code of 27_8 is found in the Test Mode register.

Subsection 24 (SUB24)

This subsection tests for the CM request/resync error.

Subsection 25 (SUB25)

This subsection tests the force mark line parity bit low.

Subsection 26 (SUB26)

This subsection tests the force tag-out/in parity error.

Subsection 27 (SUB27)

This subsection tests the force response code parity error.

Subsection 28 (SUB28)

This subsection tests the invert function-out parity bit.

Subsection 29 (SUB29)

This subsection tests the force address out parity bits low.

Subsection 30 (SUB30)

This subsection tests the parity error reporting when data-in parity is forced low on all groups.

Subsection 31 (SUB31)

This subsection tests the force data-out parity bit low.

Subsection 32 (SUB32)

This subsection tests the invert channel parity on channel 17_8, or when the parity is inverted on channel 15_8 data bus.

Subsection 33 (SUB33)

This subsection tests the invert Maintenance register write parity on the data being written into the register.

Subsection 34 (SUB34)

This subsection tests the invert nanocode parity.

Subsection 35 (SUB35)

This subsection tests the invert read parity bit.

Subsection 36 (SUB36)

This subsection tests parity error reporting when the parity is inverted on channel 15 data bus.

Subsection 37 (SUB37)

This is the test for radial interface read.

Subsection 38 (SUB38)

This is the test for radial interface write.

Subsection 39 (SUB39)

This subsection tests the force lost clock error.

Subsection 40 (SUB40)

This subsection tests the force CM request at the ADU.

Subsection 41 (SUB41)

This subsection tests the force tag-in parity error at the ADU.

Subsection 42 (SUB42)

This subsection tests the force response code parity error at the ADU.

Subsection 43 (SUB43)

This subsection tests the force D5 full at the ADU.

Subsection 44 (SUB44)

This subsection tests the force CMC busy at the CMI.

Subsection 45 (SUB45)

This subsection tests invert OS bounds address parity at the checker and OS bounds violation when slave PP tries to write CM outside allowed region specified by OS bounds.

Subsection 46 (SUB46)

This subsection tests the set inhibit PP CM request.

Subsection 47 (SUB47)

This subsection tests the clear inhibit PP CM request.

DST42 - Display Alignment Test

The Display Alignment Test (DST42) tests the interface to and from the CC545 display console. This test requires human interaction and all faults are detected by the operator.

Section Descriptions

The DST42 test is divided into eight sections.

Section	Tag	Description
00	SEC00	Tests dot function, CDC display code.
01	SEC01	Tests one character, full screen, CDC display code.
02	SEC02	Tests full alphabet, CDC display code.
03	SEC03	Tests intensity, CDC display code.
04	SEC04	Tests dot function, ASCII display code.
05	SEC05	Tests one character, full screen, ASCII display code.
06	SEC06	Tests full alphabet, ASCII display code.
07	SEC07	Tests intensity, ASCII display code.

A keyboard check may be accomplished by inputting characters to CMSE without pressing **CR**. These characters are displayed by CMSE, but are not interpreted as commands, unless the **CR** key is pressed.

Every subsection in DST42 has three conditions:

Condition	Description
00	Displays data on left screen.
01	Displays data on right screen.
02	Displays data on both screens.

Sections 00 and 04 (SEC00, SEC04)

These sections test the Dot function with CDC display code and ASCII display code, respectively. They have the following two subsections:

Subsection	Tag	Description
00	DOTTST	Displays 32-by-32 dot raster.
01	CRSTST	Displays crossed-diagonal lines.

Sections 01 and 05 (SEC01, SEC05)

These sections test the display of a full screen by filling the screen with one character. The operator enters the desired character when the message, ENTER CHARACTER is displayed. Each subsection contains the following three subsections:

Subsection	Tag	Description
00	DISCHR	Displays small characters.
01	DISCHR	Displays medium characters.
02	DISCHR	Displays large characters.

Sections 02 and 06 (SEC02, SEC06)

These sections display the full alphabet in CDC display code and ASCII display code, respectively. They have the following three subsections:

Subsection	Tag	Description
00	DISFUL	Displays small characters.
01	DISFUL	Displays medium characters.
02	DISFUL	Displays large characters.

Sections 03 and 07 (SEC03, SEC07)

These sections display the message INTENSITY ADJUST in small, medium, and large characters. The operator should attempt to adjust the intensity of the display station. These sections have only one subsection, INTENS.

CRA42 – Clock and Remote Access Test

The Clock and Remote Access Test (CRA42) tests the special features available on the two-port mux.

NOTE

CRA42 runs in standalone mode (not under CMSE) only from the CC545 display console.

Section Descriptions

CRA42 is divided into six sections.

CRA42 does not require a terminal to be connected to the ports for sections 00 through 03. Sections 04 and 05 require that a pair of modems and a terminal be connected to the port to be tested. Refer to the paragraph titled Procedures for Remote Sections of CRA42 for information about connecting the modems. The test uses even parity and two-stop bits for sections 04 and 05. These settings cannot be changed. TMP42 must run error free before CRA42 can run.

NOTE

CRA42 requires operator intervention to run sections 04 and 05.

Section	Tag	Description
00	SEC00	Reads and displays the wall clock.
01	SEC01	Tests the wall clock by writing different test patterns.
02	SEC02	Tests the wall clock increment feature.
03	SEC03	Enters the time in the wall clock.
04	SEC04	Tests auto-answer on port 0.
05	SEC05	Tests auto-answer on port 1.

Procedures for Remote Sections of CRA42

If either or both sections 04 and 05 are selected, CRA42 displays the following message after section 03:

```
AUTO ANSWER TEST
-   SET PORT-OPTION SWITCH TO
    MSG ONLY OR DS-POWER ENABLED
-   CONNECT MODEM ON
    PORT 0
-   SPACE BAR TO CONTINUE
```

Sections 04 and 05 can only be run from a remote terminal (separate from the operator console) over a telephone link as shown in figure 3-1.

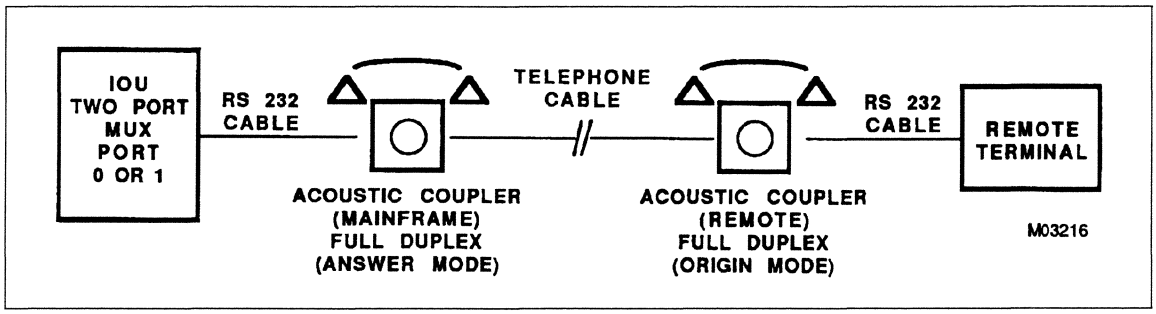


Figure 3-1. Auto-Answer Test Connections

The following procedure is used to run section 04:

1. Connect the modem to the system port (port 0 for section 04 or port 1 for section 05).

2. Connect the remote terminal to a modem.

3. Set the port-option switch on the two-port-mux box to **MSG ONLY**.

4. Press the **Space Bar** on the operator console to start the test. CRA42 monitors the Ring Indicator signal by reading the port status after you press the **Space Bar**. After 1 minute, the section is aborted.

5. Dial the telephone number of the system from the telephone connected to the remote terminal. When CRA42 detects two rings, it sets the Data Terminal Ready (DTR) and Request To Send (RTS) signals and sends the following message to the remote terminal. There is a delay of 15 seconds before this message is displayed.

   ```
   ENTER CHARACTER-UPPER CASE
   OR
   SPACE BAR TO END SECTION
   ```

   If a character is entered, it is echoed back to the remote terminal.

6. Press the **Space Bar** on the remote terminal to complete the section and advance to the next section.

   If section 05 is selected, CRA42 displays the following message after section 04.

   ```
   AUTO ANSWER TEST
   - SET PORT-OPTION SWITCH TO
     MSG ONLY OR DS-POWER ENABLED
   - CONNECT MODEM ON
     PORT 1
   - SPACE BAR TO CONTINUE
   ```

   Repeat steps 1 through 6 to run section 05.

TPM42 – Two-Port Multiplexer Test

The Two-Port Multiplexer Test (TPM42) tests the interface to and from the connected terminals. The first four sections run independently and require no operator interaction except at error stops. All other sections (05 through 08) of the test require operator interaction, and the faults detected by the test are reported on the deadstart terminal.

The test exercises only the nondeadstart ports. At least one test terminal must be connected to execute the test. If the CC634-B is connected to port 0, it tests only port 0. To test the other port, the CC634-B must be connected to the other port, or two terminals can be used, one connected to each port.

The test runs with the CC634-B set in page mode and with cursor positioning enabled (switch inside CC634-B).

NOTE

TPM42 cannot be used to test the two-port mux if a PC is connected to either port 0 or 1. The TPA42 diagnostic must be used.

The following message is displayed only at the beginning of the test when the set PARAM message is displayed:

```
SET DISPLAY TERMINAL

ON LINE,  PAGE MODE,  FULL DUP.

(LOC 141B)=PARITY SELECT

00=ODD,  01=EVEN,  02=NO PARITY

(LOC 142B)=STOP BITS

00=1 STOP BIT,  01=2  STOP BITS.

DEFAULT-EVEN PARITY/2 STOP BITS
```

When parameter word 142 has the upper bit set (1xxxxx), the two-port-mux test expects one of the ports to be connected to a modem, and the following message is displayed after section 04 has completed execution:

```
TPM4 FOUND NO CARRIER ON STATUS
THE MODEM PARAMETER IS SELECTED.
DIAL PHONE FOR CONNECT TO TWO PORT MUX
TEST WILL WAIT FOR CARRIER ON STATUS
```

Section Descriptions

The TPM42 is divided into the following nine sections:

Section	Tag	Description
00	SEC00	Tests function response and FIFO.
01	SEC01	Tests the loopback with 5-, 6-, 7-, and 8-bit mode (8-bit only when connected to a PC console).
02	SEC02	Tests data in overrun.
03	SEC03	Tests the DTR function and connect function.
04	SEC04	Tests Master-Clear and Disconnect functions.
05	SEC05	Tests X-Y positioning, displays dot code, and parity-error detection.
06	SEC06	Tests display full alphabet.
07	SEC07	Tests X-Y positioning by displaying cross-diagonal lines.
08	SEC08	Tests the read deadstart port, Terminal function.

Section 00 (SEC00)

This section has the following four subsections:

Subsection	Tag	Description
00	SUBSEC00	Checks to see if the test initialization detected a response to the DTR function.
01	SUBSEC01	Checks for the response on issued functions.
02	SUBSEC02	Checks for FIFO operation.
03	SUBSEC03	Checks if the DSR signal remains set if previously set by the DTR function after the FIFO has emptied.

Section 01 (SEC01)

This section has the following four subsections:

Subsection	Tag	Description
00	SUBSEC00	Tests the loopback feature with 8-bit mode.
01	SUBSEC01	Tests the loopback feature with 7-bit mode.
02	SUBSEC02	Tests the loopback feature with 6-bit mode.
03	SUBSEC03	Tests the loopback feature with 5-bit mode.

Section 02 (SEC02)

This section tests the data in overrun bit.

Each subsection has two conditions. Condition 00 tests port 0 (if connected) and condition 01 tests port 1 (if connected).

Section 03 (SEC03)

This section has the following four subsections:

Subsection	Tag	Description
00	SUBSEC00	Tests the Data-Terminal-Ready function (40X).
01	SUBSEC01	Same as subsection 00.
02	SUBSEC02	Tests the Connect function.
03	SUBSEC03	Verifies that the TPM will not clear the Data Terminal Ready signal after the FIF0 has emptied if it was set previously by the set-data-terminal-ready function.

Section 04 (SEC04)

This section has the following two subsections:

Subsection	Tag	Description
00	SUBSEC00	Tests the Master-Clear function (700) and verifies that the input and output buffers are cleared with this function.
01	SUBSEC01	Tests the Disconnect function (6000). Checks that the two-port mux will continue to output after the disconnect when data is still in the buffer.

Section 05 (SEC05)

This section has the following two subsections:

Subsection	Tag	Description
00	SUBSEC00	This subsection displays the message, ENTER CHARACTER. The test checks to see if characters can be entered while the two-port mux is outputting.
01	SUBSEC01	Tests the parity-error detection and displays a full screen of the character. The character is entered by the operator when the ENTER CHARACTER message is displayed.

Section 06 (SEC06)

This section displays the full alphabet.

Section 07 (SEC07)

This section tests X-Y positioning using cross-diagonal lines.

Section 08 (SEC08)

This section tests the Read Deadstart Port and Terminal function. The deadstart port and terminal are displayed on the console and operater input is required to continue the test.

TPA42 - Two-Port Multiplexer Test

The Two-Port Multiplexer test (TPA42) is a standalone only test.

TPA42 tests the interface between the PP using channel 15_8 and port 0 or 1. The test requires user interaction and all faults are detected by the operator when the test stops with a display message error or a hung condition. The test stops in a hung state only if a function code failure occurs.

The test exercises ports 0 and 1 when a CC545 is used as the main console. Only port 1 is exercised when the PC is the main console.

The read and write clock sections are run from both ports and therefore test all ports.

NOTE

If sections 07 and 08 are run, it is necessary to reset the wall clock before the operating system is loaded because of a mismatch between the two-port-mux clock and the last entry made in the PC console clock monitor. This mismatch may cause software update problems.

The test is run with the quicklook bit cleared so that the operator can see the subsection and condition counters incrementing. Otherwise, too much time is wasted determining the state of the PP, running or not running.

The following message is displayed only at the beginning of the test when the set PARAM message is displayed:

```
SET DISPLAY TERMINAL

ON LINE,  PAGE MODE,  FULL DUP.

(LOC 141B)=PARITY SELECT

00=ODD,  01=EVEN,  02=NO PARITY

(LOC 142B)=STOP BITS

00=1 STOP BIT,  01=2  STOP BITS.

DEFAULT-EVEN PARITY/2 STOP BITS
```

Section Descriptions

The TPA42 test is divided into ten sections:

Section	Tag	Description
00	SEC00	Tests the Read Deadstart Port and Terminal function.
01	SEC01	Tests the function response and FIFO.
02	SEC02	Tests the DTR and Connect function.
03	SEC03	Tests the loopback with 5-, 6-, 7-, and 8-bit mode.
04	SEC04	Tests the Master Clear and Disconnect functions.
05	SEC05	Tests the data in overrun.
06	SEC06	Reads and displays the wall clock.
07	SEC07	Tests the wall clock by writing different test patterns.
08	SEC08	Tests the wall clock increment feature.
09	SEC09	Enters the time in the wall clock.

Section 00 (SEC00)

This section tests the Read Deadstart Port and Terminal function. The deadstart port and terminal are displayed on the console and operator input is required to continue the test. If the CC545 is the monitor console, the Read Deadstart function is executed on both ports and displayed.

Section 01 (SEC01)

This section has the following four subsections:

Subsection	Tag	Description
00	SUBSEC00	Checks to see if the test initialization detected a response to the DTR function.
01	SUBSEC01	Checks for the response on issued functions.
02	SUBSEC02	Checks the FIFO operation.
03	SUBSEC03	Checks if the DTR signal remains set if previously set by the DTR function after the FIFO has emptied.

Section 02 (SEC02)

This section has the following four subsections:

Subsection	Tag	Description
00	SUBSEC00	Tests the loopback feature with 8-bit mode.
01	SUBSEC01	Tests the loopback feature with 7-bit mode.
02	SUBSEC02	Tests the loopback feature with 6-bit mode.
03	SUBSEC03	Tests the loopback feature with 5-bit mode.

Section 03 (SEC03)

Each section has two conditions; condition 00 tests port 0, while condition 01 tests port 1. Section 03 has the following two subsections:

Subsection	Tag	Description
00	SUBSEC00	Tests the four word input buffer.
01	SUBSEC01	Tests the data overrun bit.

Section 04 (SEC04)

Each section has two conditions; condition 00 tests port 0, while condition 01 tests port 1. Section 04 has the following four sections:

Subsection	Tag	Description
00	SUBSEC00	Tests the Clear Data Terminal Ready function (400).
01	SUBSEC01	Tests the Set Data Terminal Ready function (401).
02	SUBSEC02	Tests the Connect function (700x).
03	SUBSEC03	Tests the two-port mux. Does not clear the Data Terminal Ready signal after the FIFO has emptied if it was set previously by the Set Data Terminal Ready function.

Section 05 (SEC05)

Each section has two conditions; condition 00 tests port 0, while condition 01 tests port 1. Section 05 has the following two subsections:

Subsection	Tag	Description
00	SUBSEC00	Tests the Master-Clear function (700) and verifies that the input and output buffers are cleared with this function.
01	SUBSEC01	Tests the Disconnect function (6000). Checks that the two-port mux continues to output after the disconnect when data is still in the buffer.

Section 06 (SEC06)

Section 06 has the following subsection:

Subsection	Tag	Description
00	SUBSEC00	Reads calendar clock and displays it. Each condition is a different read as it switches ports each read.

Section 07 (SEC07)

This section tests the wall clock by writing a pattern of data to clock then reading it back to verify. There are five fields defining the year (YY), month (MM), day (DD), hour (HH), and minute (MN) as shown in the following example.:

```
:---------------------------------:
: 00 : YY : MM : DD : HH : MN : 00 :
:    :    :    :    :    :    :    :
Field      Condition        Patterns
*****      *********        ********
 YY           0               00
 YY           1               01
  :           :                :
 YY          98               98
 YY          99               99

 MM         100               01
 MM         101               02
  :           :                :
 MM         110               11
 MM         111               12

 DD         112               01
 DD         113               02
  :           :                :
 DD         141               30
 DD         142               31

 HH         143               00
 HH         144               01
  :           :                :
 HH         165               22
 HH         166               23

 MN         167               00
 MN         168               01
  :           :                :
 MN         225               58
 MN         226               59
```

In each condition, a pattern is written in a field; other fields contain 01. For example:

Condition 03: 00 03 01 01 01 01 00
Condition 98: 00 98 01 01 01 01 00
Condition 108: 00 01 09 01 01 01 00.

Section 08 (SEC08)

This section has the following three subsections:

Subsection	Tag	Description
00	SUBXER	This subsection checks propagation delay from minute-to-hour to day-to-month and to-year.

NOTE

To run subsections 1 and 2, clear quicklook mode bit.

01	SUBONE	This subsection checks the propagation of the carry (the increment of the clock). minute → hour, hour → day, day → month, month → year.
02	SUBTWO	This subsection tests the calendar clock to make sure that the last day is right for each month.

Section 09 (SEC09)

This section has the following subsection.

SUBXER — This section can be used to set date and time in the wall clock from the keyboard. When the message ENTER DATE/TIME YYMMDDHHMN is displayed, enter data through the keyboard in the above form, where YY = year, MM = month, DD = day, HH = hour, MN = min. No separators are required. For example: 8506100812. If data is illegal an INVALID ENTRY message is displayed. If good data is entered, it is written to the clock, read back, and displayed. Type ABS **CR** to exit or abort this section at the end of the test.

Isolation Diagnostics

The isolation diagnostic (FII42) analyzes errors reported by the detection tests and attempts to identify a group of logic paks that are likely to be causing the problem.

FII42 – Fault Isolation Program

The Fault Isolation Program (FII42) is a table-driven analyzer that uses data recorded by the IOUCP2 Error Processing Routine (ERRCP) during IOU test execution.

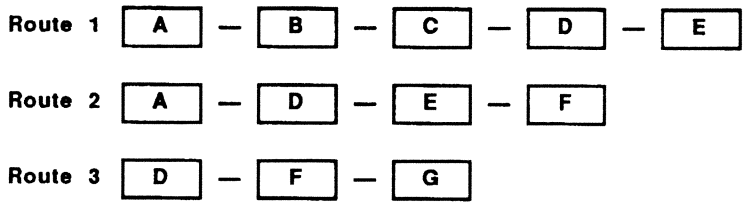
ERRCP records three classes of errors and assigns route numbers to them:

Error Class	Route Numbers
Unexpected parity errors	0
Invert parity errors	1 through 77_8
Data/control errors	100 through 177_8

FII42 analyzes the errors recorded by the detection tests using the route numbers as the base for isolation.

The route numbers for fault isolation are an index to a predefined string of modules.

Example:



M03217

FII42 sets error location (ERROR) to 20 and starts by processing route number 0. If an error is found in this route number, location (ERROR) is decremented by one for each pak that is isolated. If no errors are found for route number 0, FII42 processes route numbers 100 through 177_8. If no errors are found for route numbers 100 through 177_8, FII42 processes errors for route numbers 1 through 77_8.

When FII42 finds errors in the route numbers, it calls subroutines to find and identify the pak associated with the highest number of error routes. The subroutine assigns priority number one to paks with the highest count and then identifies lower priority paks. The maximum number of paks that can be called is 10.

Log Errors Routine

This routine is built into the IOUCP2 (IOU Control Program). It is used to log errors in the error buffer for fault isolation. It is called by IOUCP2 Error Processing Routine (ERRCP). The following classes of errors are logged:

- Unexpected parity error: When a parity error is detected, IOUCP2 generates a ROUTE=0 code and passes the contents of Fault Status registers 1 and 2 (FS1 and FS2) in (PEBUF) to the Error Log Routine.
- Invert parity error: When MRT42 detects an error, it generates a ROUTE=1 through 77_8 code and passes the logical difference between expected and received FS1 and FS2 in PEBUF to the Error Log Routine. The following parameters are also passed to the Error Log Routine.

  (EXPAT) Expected Pattern Buffer

  (RCPAT) Received Pattern Buffer

  (PPNUM) Failing PP Number

  (CHNUM) Failing CH Number

  (PEBUF) Parity Error Buffer

- Data/control error: When tests detect an error, they generate a ROUTE=100 through 177_8 code and pass the following parameters to the Error Log Routine.

  (EXPAT) Expected Pattern Buffer

  (RCPAT) Received Pattern Buffer

  (PPNUM) Failing PP Number

  (CHNUM) Failing CH Number

  (PEBUF) Parity Error Buffer

Figure 3-2 shows the Monitor PP map. The errors are recorded by Error Log Routine in the Error Log Buffer and processed by FII42.

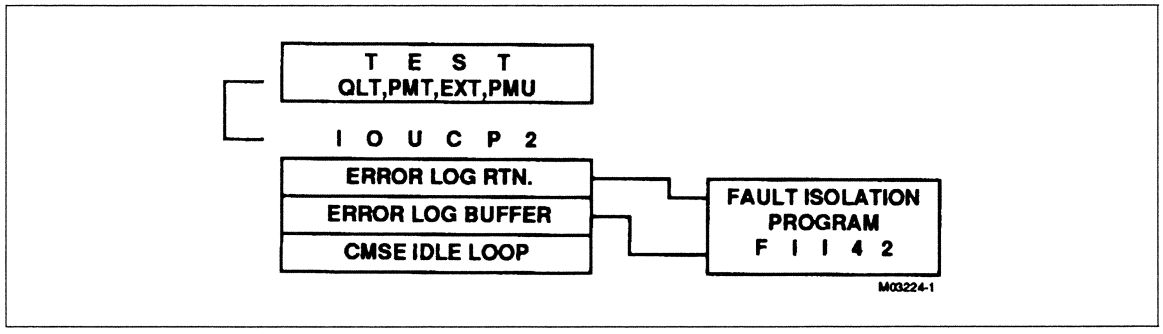


Figure 3-2. Monitor PP Map

Unexpected Parity Error Recording

The route number=0 is transmitted whenever unexpected parity errors occur. As shown in figure 3-3, the errors are logged by setting the RPEBUF equal to the accumulative logical difference of FS1 and FS2 from PEBUF, and PCOUNTER is incremented by one.

The size of the Recorded Parity Error Buffer (RPEBUF) is 20_8 locations.

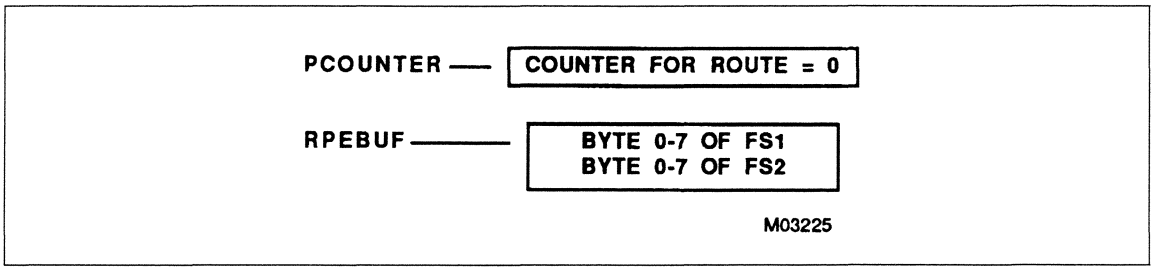


Figure 3-3. Parity Error Buffer

Invert Parity Error Recording

The routes numbered 1 through 77_8 are the invert parity error routes. When one of these errors is reported, the corresponding location (PCOUNTER+(ROUTE)) is incremented by one.

Also, a cumulative logical difference of FS1 and FS2 is recorded in location RPEBUF+20_8 to location RPEBUF+37_8.

Figure 3-4 shows what these locations represent.

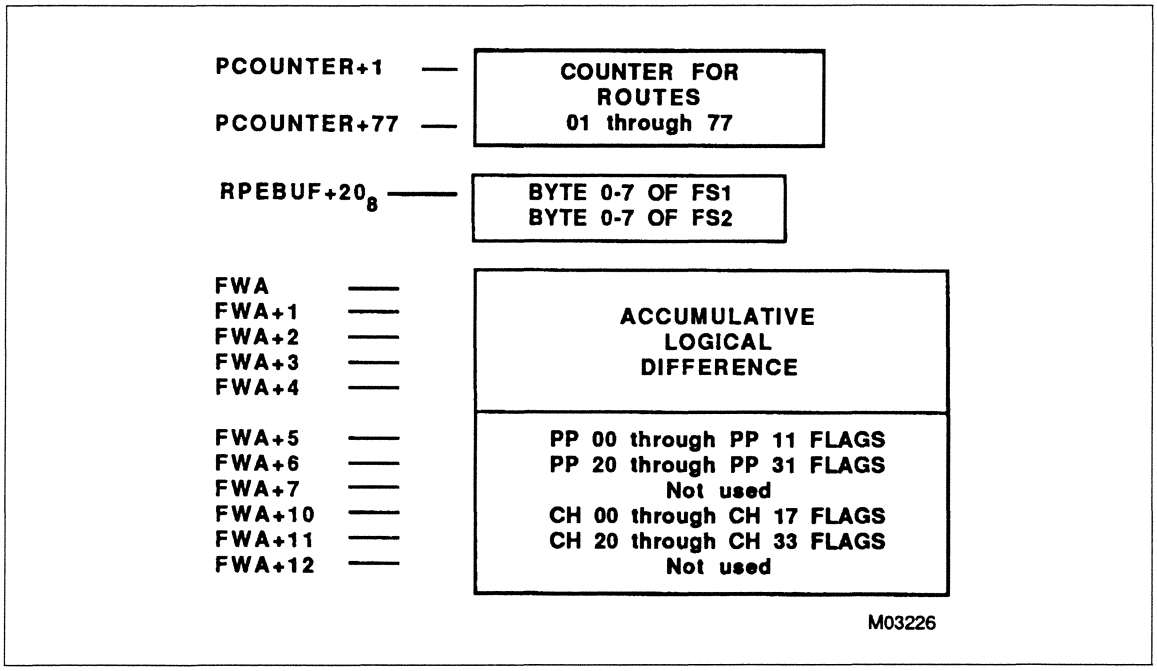


Figure 3-4. Invert Parity Error Buffer

Control and Data Error Recording

The routes numbered 100 through 177_8 represent the errors for data and control flow. The route numbers are used to generate the starting address for cumulative logical difference, PP, and channel flags when an error is detected.

The logical difference is calculated using expected (EXPAT) and received (RECPAT) patterns. The PP and channel position flags are calculated using (PPNUM) and (CHNUM).

The corresponding counter for failing route (PCOUNTER+(ROUTE)) is incremented by one.

Figure 3-5 shows the buffer for one failing route number.

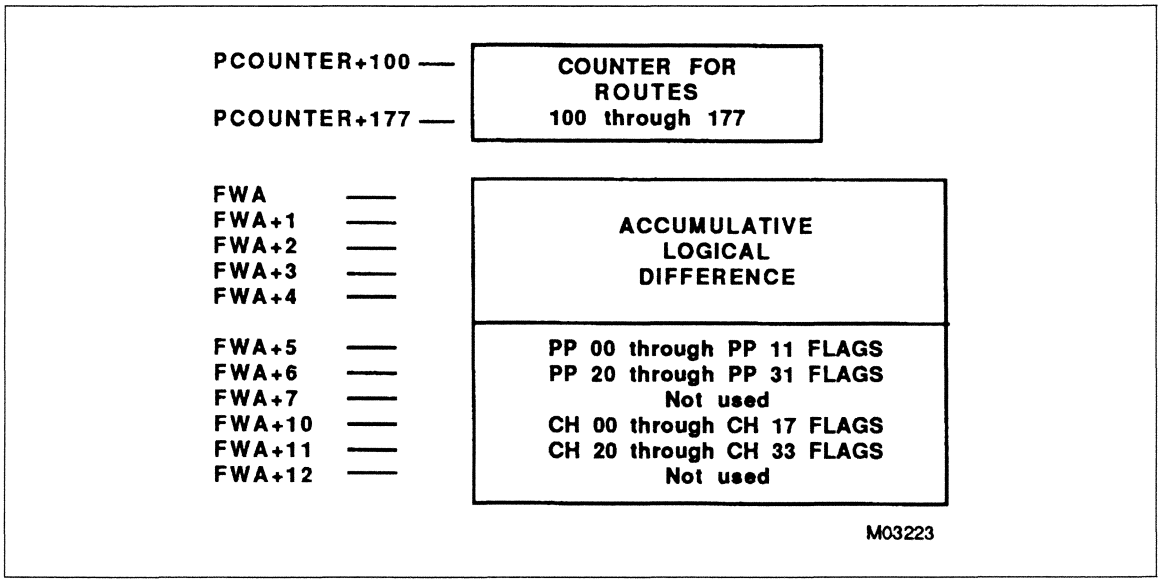


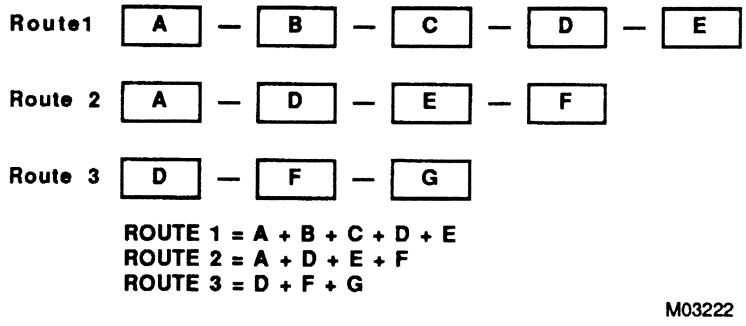
Figure 3-5. Control and Data Error

Fault Isolation Program

FII42 analyzes the errors recorded by the detection tests using route numbers as a base for isolation.

The route numbers for FII42 are considered as a index for the string of the involved modules.

Example:



FII42 sets location (ERROR) to 20_{10}, which is used as a flag when an error condition is detected.

The routine starts the processing of the unexpected parity error route (Route=0). If the unexpected parity error route is nonzero, location (ERROR) is decremented and the suspected failing paks are isolated. If location (ERROR) is equal to 20_{10}, route numbers 100 through 177_8 are processed. These route numbers represent the data and control flow errors. This processing is done by checking the corresponding location in the counter buffer for nonzero. When a location is found to be nonzero, location (ERROR) is decremented and the suspected failing paks are isolated.

If location (ERROR) is still equal to 20_{10}, the invert parity routes are processed. These routes represent the MRT42 detected errors. If there are no errors for these routes, FII42 calls CMSE to display the message NO ERROR RECORDED.

The FISORT routine is called if the unexpected parity error route or any of the data and control routes are nonzero. The FISORT route will find the pak with the highest number of calls. The counter for that pak has the highest count.

The FISOL routine is then called to set priority and identify the pak type and location. The calling table is passed to CMSE to display SUSPECTED FAILING MODULE/S (FISM1, Fault Isolation Message 1).

Parity Error Processing

This routine checks if location PCOUNTER is zero and, if it is, returns control to FICP with location (ERROR) unchanged. If PCOUNTER is nonzero, error location (ERROR) is decremented and the COPYPE routine is called to copy the recorded accumulative logical difference for this route in PEBUF and:

- Set (NUMBAR) number of failing barrels.
- Set (BARNUM) failing barrel number.

The parity error processor then increments the involved paks. If no error is found, it processes routes 100 through 177_8.

Data and Control Processing

The errors with route numbers 100_8 through 177_8 are processed in the following manner:

- Use COPYER routine to set the following:
  - Accumulative logical difference to received buffer.
  - Failing PP flags in PP parameter area.
  - Failing channel flags in channel parameter area.
  - Number of failing barrels (NUMBAR).
  - Failing physical barrel number (BARNUM).
- Call the processor for that route number to:
  - Determine if it is a data or control failure.
  - Find the number of failing bits and failing bit numbers.
  - Find the failing channel number and number of failing channels if it is channel failure.
  - Increment counters for the paks that caused the error.

Invert Parity Processing

This routine sets location (ERROR) to zero and checks if any location in the PCOUNTER buffer is nonzero and, if not, returns control to the Fault Isolation Control Program (FICP) with location (ERROR) unchanged. It sets location (ERROR) to nonzero if any failing route is found. If there is any error, it calls COPYPE routine to copy the recorded accumulative logical difference to PEBUF and:

- Set NUMBAR.
- Set BARNUM.

The routine then builds the display message for the suspected failing pak(s), which is displayed on the console.

Module Sorting

The Pak Sorting Routine (FISORT) is used to find the pak counter with the highest count and set Maximum Count Value (MAXCNT).

If the highest count is zero (processors were not able to determine the failing pak multiple uncorrelated error), the FISM2 (Fault Isolation Message 2) is passed to CMSE.

Fault Isolation

The Fault Isolation Routine (FISOL) is called to identify the pak type and pak location and to set the priority for the paks that have counters nonzero.

FISOL assigns priority number one to all the paks with the highest counters. It then decrements the maximum count and increments the priority number for the next set of paks with the same count as the current maximum count.

FISOL calls MODID for the pak location.

The maximum number of paks to be called is 10.

CYBER 2000 IOU and Associated Equipment 4

CYBER 2000 IOU and Associated Equipment 4

LDX – Long Deadstart Sequence

The Long Deadstart Sequence (LDX) is a quick look program that resides in read-only memory (ROM) attached to upper core of logical PP00 memory (5400 through 7777_8). LDX transfers to barrel 1 after barrel 0 completes. LDX continues to transfer to barrels 2 and 3 if they are installed. Peripheral processors (PPs) within each barrel may be reconfigured by using the deadstart display and entering the RP x command. Initiating the long deadstart sequence sets the following conditions.

- A master clear is issued to all PPs and all error status bits are cleared.
- PP00 is forced to read the test program stored in ROM.
- PP00 address is forced to 6000_8 (5400 through 5700_8 is for test update and transfer routine).
- A PP register display is brought up on the right-hand side of the screen.

While LDX is executing, certain error status bits are monitored to determine if the hardware has been detected. If a hardware failure is detected, a message is displayed below the PP register.

Each error stop is indicated by a unique P register address. The customer engineer (CE) determines what error condition was detected by examining the P register display and referring to the LDX error tables. Refer to the CYBER 2000 System Troubleshooting Guide, listed in About This Manual.

When execution of LDX is finished (all installed barrels run), the short deadstart sequence is initiated using the program currently displayed on the screen.

LDX Tests

LDX performs the following sequence of tests:

1. Arithmetic Unit Test
2. PP00 Memory Test (up to location 5377_8)
3. One Word Channel Test
4. Block I/O Transfer Test
5. Block I/O Transfer with Conversion Test
6. Multi-PP Arithmetic Unit Test
7. LDX Bit Test
8. Detection Tests

The following paragraphs describe these tests.

Arithmetic Unit Test

The tests performed by the Arithmetic Unit Test are:

Test	Description
Unconditional Jump Test	Checks execution of the 03 instruction and the lower 6 bits of the d portion of the instruction. Error stops are: - on 0300 or 0377 instruction for single Q-bit failure. - indicated by hardware timeout counter error for a multiple Q-bit failure or P register failure.
Conditional Jump Test	Checks instructions 04 through 07 with A equal to 0, A equal to 1, and A equal to minus 0. Error stops are on 03 through 07 instructions.
1x Instructions Test	Checks instructions 10 through 17 using the lower 6 bits of the A register. Error stops are on branch instructions.
2x Instructions Test	Checks instructions 20 through 25, arithmetic operations in the A register with all 18 bits, and shift network. Error stops are on branch instructions.
3x Instructions Test	Checks instructions 30 through 37 using the lower addresses of PP00 memory for store instructions. Error stops are on branch instructions.
4x Instructions Test	Checks instructions 40 through 47 and all 16 bits of the Q register. Lower addresses of PP00 memory are used to check restore capability. Error stops are on branch instructions.
5x Instructions Test	Checks instructions 50 through 57 with and without the Q flag set. Error stops are on branch instructions.
13x Instructions Test	Checks execution of instructions 130 through 137 with 16-bit operands. Error stops on branch instructions.
14x Instructions Test	Checks execution of instructions 140 through 147 with 16-bit operands. Error stops are on branch instructions.
15x Instructions Test	Checks execution of instructions 150 through 157 with 16-bit operands, with and without the Q flag set. Error stops are on branch instructions.
RJM and LJM Instructions	Checks execution of instructions. Test 01 and 02 using previously written addresses in lower core of PP00 memory. Error stops are on branch instructions.

PP00 Memory Test

This test executes from low addresses of PP memory. The test is transferred from ROMs to PP memory and checked for correct transfer. Error stop is on a branch instruction (PP00 at location 24_8, PP01 through PP04 at PP00 memory location 210_8 plus PP number).

The test then checks PP memory with 10 fixed data patterns and addresses. Error stop is on a branch instruction. The contents of the A register represent the failing bits. The failing pattern and failing address are stored in low addresses of PP00 memory. The following table indicates the bit locations in PP00 memory.

PPM00 Location	Contents
34_8	Failing bits
35_8	Failing patterns
36_8	Failing addresses

One Word Channel Test

This test checks channels with one word input/output (I/O) using all possible bit combinations and Channel status during I/O. Barrel 0 tests channels 0, 1, and 13. All other barrels test channels 12, 15, and 17. The remaining channels are not tested because they may not be installed. Error stops are on branch instructions and are unique for each Channel status and data error.

Block I/O Transfer Test

This test checks inter-PP communication channel 12 using a slightly modified PP memory test. A timeout counter protects PP00 from a hung condition. Error stops are on branch instructions that are unique for each channel/PP.

Block I/O Transfer with Conversion Test

This test checks inter-PP communication with 12-to-16-bit and 16-to-12-bit conversion over channel 12. Error stops are on branch instructions. A timeout feature protects PP00 from a hung condition.

Multi-PP Arithmetic Unit Test

This test is the same as the Arithmetic Unit Test described previously, but runs in PP01 through PP04 simultaneously. If an error is detected, the PP in which the test is running stops and P register stop lights are the same as for the Arithmetic Unit Test. PP00 is in a loop waiting for a message from PPxx indicating that the test successfully passed. The looping address is unique to each PP. Each PP is force deadstarted, waits for the channel 13 flag to set before starting, and sets the channel 12 flag when completed.

LDX Bit Test

This test checks the LDX bit in the Maintenance register. If the bit is clear, PP00 stops and the P register indicates an error in the Maintenance register.

Detection Tests

The detection tests are a series of programs that test each PP and channel in the CYBER 2000 IOU (IOU4CE) for correct functional operation. These tests also record errors such as differences between expected and actual (received) results and generate internal codes that identify the area of the CYBER 2000 IOU (IOU4CE) under test when the error was detected.

QLT46 – Quick Look Test

The Quick Look Test tests one-word I/O over each channel. The test uses hardware features such as force idle PP and force deadstart PP.

At the end of the test, all tested channels are set active, empty, and all associated PPs, if selected, are in an idle condition.

Section Descriptions

The Quick Look Test section numbers, converted to octal, represent the channel under test (that is, section 8 represents channel 10_8). For a channel to be tested, both the section and the channel must be selected by bits in the section select and channel flag parameter words. Refer to parameters and control words in the MSL 153/155 IOU Test Procedures Reference manual listed in the preface of this manual.

This section has the following four subsections.

Subsection	Tag	Description
00	SUBSEC00	Initializes the associated PP (Force idle PP).
01	SUBSEC01	Tests Channel Flags status for set and clear.
02	SUBSEC02	Tests one word pattern over each channel selected.
03	SUBSEC03	Checks negative testing.

At the end of the section, channel status is active, empty, and the associated PP is in an idle condition.

Subsection 00 (SUBSEC00)

Initialize channel number for all instruction in the slave PP. Deadstart PP and check K register for 107700 idle condition.

Subsection 01 (SUBSEC01)

Test all channel flags for set/clear.

Subsection 02 (SUBSEC02)

Subsection 02 tests one word pattern over each channel selected to be tested, and checks the status of the channel during the I/O transfer. This subsection has 1000_8 conditions that use different patterns and complements of the pattern. There are 152_8 fixed patterns and 626_8 randomly generated patterns.

Each condition uses one pattern and the complement of the pattern. During this I/O, the test checks for correct channel status and channel error flag clear status.

Subsection 03 (SUBSEC03)

Subsection 03 has the following ten conditions.

Condition	Description
00	Disconnects already inactive on channel with escape bit set.
01	Outputs one word on inactive channel with escape bit set.
02	Inputs one word from inactive channel with escape bit set.
03	Blocks input over inactive channel expecting that the first word address of the block input is cleared.
04	Blocks output over inactive channel expecting that the contents of the A register are not changed.
05	Activates already active channel with escape bit set.
06	Functions FAN on active channel with escape bit set.
07	Functions FNC on active channel with escape bit set.
08	Forces channel active and full and checks for empty when channel is deactivated.
09	Checks channel for empty and active signals.

NOTE

It is possible for subsection 03 to cause the PP from which the test is being executed to hang if the escape feature (bit 2^5 of I/O instructions) does not work. There is no way to prevent this hang if the feature fails.

If all channels for the current barrel are turned off, channel 17_8 is used as a default test channel.

PMT46 - PP Memory Test 1

The PP Memory Test 1 (PMT46) checks the operation of the PP memories except for the PP in which the PMT46 test resides. The test does not rely on code executing in the PP memory being tested. PP memories are tested sequentially using force deadstart, idle, and force dump. Since the PP in which PMT46 resides was checked by the LDX test, PMT46 assumes that the PP is operational.

Section Descriptions

The PP Memory Test 1 section numbers, in decimal, represent the PP memory being tested. For a memory to be tested, both the section and the PP must be selected by bits in the section select and PP flag parameter words. Refer to parameters and control words in the MSL 153/155 IOU Test Procedures Reference manual listed in the preface of this manual.

All data transfers for PPs are done over channel 13_8.

This section has 32 subsections numbered 00 through 31, as described in the following table.

Subsection	Tag	Description
00	B00.00	Forces deadstart PP with check.
01	B01.00	Checks P and A registers.
02 to 30	B02.00	Tests memory with patterns.
31	B03.00	Tests memory with address patterns.

At the beginning of the test, all PPs are forced into a hung condition by a force idle feature. The PPs are left idle at the end of each section and are in an idle condition at the end of the test.

Subsection 00 (B00.00)

This subsection has the following five conditions.

Condition	Description
00	Forces PP to deadstart condition.
01	Checks P register of the PP being tested for the value 177777_8.
02	Checks Q register of the PP being tested for deadstart channel value.
03	Checks K register of the PP being tested for the value 007100_8.
04	Checks A register of the PP being tested for the value 020000_8.

Subsection 01 (B01.00)

This subsection has the following condition.

Condition	Description
00	Repeat one word output to fill tested memory. Check the P and A registers of the PP being tested for the correct count after each word of output.

Subsections 02 through 15 (B02.00)

These subsections output standard test patterns to complement every second word and then dump the test PP over the used channel to the monitor PP to check the data.

These subsections have the following two conditions.

Condition	Description
00	Outputs data blocks of 100_8 words to fill up the test PP memory.
01	Forces a dump of the test PP and inputs one word at a time, checking data.

Subsections 16 through 30 (B02.00)

These subsections use the same code as subsections 02 through 15, but use random number patterns.

Subsection 31 (B03.00)

This subsection outputs the address of the corresponding memory location to fill the test PP memory and then dumps the test PP over the used channel to the monitor PP to check the data.

This subsection has the following two conditions.

Condition	Description
00	Repeats one word output to fill the memory of the test PP.
01	Forces a dump of the test PP and inputs one word at a time, checking data.

EXT46 – Execution Unit Test

The Execution Unit Test checks instruction execution and the arithmetic units in all PPs, except the one in which the Input/Output Control Program (IOUCP) and CMSE are running. CMSE loads EXT46 into the same PP with IOUCP and distributes the updated copies to the PPs to be tested. The test is started by deactivating the communication channel for the assigned PP.

Section Descriptions

EXT46 is divided into the following three sections.

Section	Tag	Description
00	UJNTST	Tests the arithmetic unit and execution of instructions 00 through 57_8 (12-bit operand instructions).
01	TEST13X	Tests the arithmetic unit and execution of instructions 1030 through 1057_8 (16-bit operand instructions).
02	STSTEST	Tests the execution of the channel instructions.

Section 00 (UJNTST)

This section tests the 12-bit operand instructions 00 through 57_8. The test is divided into the following six subsections.

Subsection	Tag	Description
00	UJNTST	Tests execution of 00 through 07 instructions.
01	TEST1X0	Tests execution of 10 through 17 instructions, and the A register bits 2^0 through 2^5.
02	TEST2X0	Tests execution of 20 through 23 instructions, and the A register bits 2^0 through 2^{17}.
03	TEST3X	Tests execution of 30 through 37 instructions, and restore ability.
04	TEST4X	Tests execution of 40 through 47 instructions, and the A register bits 2^0 through 2^{11}.
05	TEST5X	Tests execution of 50 through 57 instructions, and the Q adder.

Section 01 (TEST13X)

This section tests the 16-bit operand instructions. The test is divided into the following four subsections.

Subsection	Tag	Description
00	TEST13X	Tests execution of 1030 through 1037 instructions and restore ability on 16-bit boundary.
01	TEST14X	Tests execution of 1040 through 1047 instructions and restore ability using indirect addressing.
02	TEST15X	Tests execution of 1050 through 1057 instructions using index addressing.
03	JMPTEST	Tests execution of 01 and 02 instructions.

Section 02 (STSTEST)

This section tests the I/O instructions (channel instructions) and has the following four subsections.

Subsection	Tag	Description
00	STSTEST	Tests 64 through 67 instructions.
01	IANTEST	Tests 70, 72, 74, and 75 instructions.
02	IOTEST	Tests 1064 through 1067 instructions.
03	ESCTEST	Tests channel instructions with escape bit (Q bit 2^5 set).

PMU46 - PP Memory Test 2

The PP Memory Test 2 (PMU46) performs a more rigorous check of the PP memories than PMT46. The advantage over PMT46 is that PMU46 loads a copy of the PP memory test routine into all PP memories that are to be checked. The PP memory test routine is then executed by the PPs. This provides more variety in the way memory is referenced and allows more strenuous code to be used.

Section Descriptions

The section, subsection, and condition structure of PMU46 is defined in reference to where the code executes in a particular test PP. The code in the test PP is executable from either the lower or upper half of the first 4K of PP memory, so that the entire memory can be tested.

Selection of sections 00 and 01 tests the entire memory.

Section	Description
00	Executes in the lower half of memory and tests all memory except test core and direct cells used by the lower half copy.
01	Executes in the upper half of memory and tests direct cells and core used by the lower half copy.

Each section includes all of the following 48 subsections except as noted.

Subsection	Tag	Description
00	SS00	Checks to find a solid addressing failure. This is not affected by the memory bounds of a section.
01 through 36	SS01 through 36	Checks standard and random patterns of available test memory.
37 through 39	SS37 through 39	Tests available test memory with a random word in each location.
40 through 46	SS40 through 46	Checks addressing by rapidly loading random addresses in a pre-arranged sequence.
47	SS47	Checks if faults are induced when address lines are selectively toggled. This is not affected by the memory bounds of a section.

Subsection 00

This subsection uses the following two conditions to check for solid address failure.

Condition	Description
00	Initializes location 0 with all zeros, and stores ones in all remaining locations with one-bit-set addresses. A nonzero word in location 0 indicates a shorted address line.
01	Initializes location 7777_8 with all ones, and stores zeros in all remaining locations with one-bit-clear addresses. A zero-bit in location 7777_8 indicates an open address line.

Subsections 01 through 06

These subsections use a standard fixed pattern. The testable memory is checked in groups of five words with the pattern, then its complement, being loaded into alternate locations. The words are quickly read back and checked in sequential order.

Subsections 07 through 14

These subsections are the same as subsections 01 through 06, except the standard patterns are circularly shifted. These sliding patterns use 16 conditions, one for each circular shift.

Subsections 15 through 21

These subsections are exactly the same as subsections 01 through 06. The placement allows staggering of fixed and sliding pattern checks.

Subsections 22 through 36

These subsections are the same as subsections 01 through 06, except the patterns are randomly chosen before the tests are copied to the test memories.

Subsections 37 through 46

These subsections have one condition. Each subsection uses a real-time-clock input to generate a series of 1000_8 addresses. For each address the previous address is used as data. This daisy chain is then rapidly retraced using load instructions. If no error is generated by rapid changes of the address/data lines, the 1000th load is the reference point.

Repeating the condition uses the same seed for the random number generation; repeating the subsection uses a new seed for each cycle.

The subsections are duplicated to allow multiple cycles without selecting a repeat subsection.

Subsection 47

This subsection checks if the address lines are affected by crosstalk. An instruction sequence is executed that causes the address lines to switch from all zeros to all ones, except for one bit to be checked for staying zero. The sequence is repeated for all 13 address bits, and then another sequence is used for checking all ones toggling to all zeros (but one).

This subsection uses the following two conditions.

Condition	Description
00	Toggles all address lines, except the test bit, from zeros to ones by executing a store with indirect addressing through location 0 that contains a one-bit-clear address.
01	Toggles all address lines, except the test bit, from ones to zeros by executing an indexed long jump instruction, from location 7776_8 to location 0 plus a one-bit-set address.

CHD46 – Channel Test

The Channel Test (CHD46) checks inter-PP data transfer over each tested channel.

The test uses a pair of PPs for inter-PP transfer. One PP is called the PP transmitter and the other, the PP receiver. The PP transmitter communicates through the Double PP Driver (DPPD) with the Input/Output IOU Control Program (IOUCP) over the communicating channel. The PP receiver communicates with the PP transmitter over the testing channel.

The PP transmitter and the PP receiver are randomly selected at the beginning of each section and whenever the test is repeated.

The execution of the complete test with one channel is considered a section. The number of selected sections specifies the number of the channels to be tested for randomly selected PP pair (default is 32 sections selected).

When all selected sections are completed, the end of the test is reported and, if repeat test is selected, new pairs are randomly generated.

The maximum number of words (size of the I/O block) to be transferred is 1000_8.

Section Descriptions

In this test, a section is the execution of the complete program using one pair of PPs that are testing one channel. The PP receiver returns received data over the channel being tested to the PP transmitter. The PP-transmitter section is divided into the following five subsections.

Subsection	Tag	Description
00	SUBSEC00	Tests one word I/O over the channel being tested using the OAN instruction.
01	SUBSEC01	Tests one word I/O over the channel being tested using the OAM instruction.
02	SUBSEC02	Tests the channel flag operation (PP-receiver number is the channel number for the channel flag test).
03	SUBSEC03	Tests block I/O over the channel being tested using the OAM instruction.
04	SUBSEC04	Tests block I/O over the channel being tested using the OAPM instruction (conversion).

Subsection 00 (SUBSEC00)

Subsection 00 tests the data and Channel status during one word I/O using the single word output (OAN) instruction over the channel being tested. The data used for I/O are word counts. Every condition is represented by one word count.

Subsection 01 (SUBSEC01)

Subsection 01 tests the data and Channel status during one word I/O using the output block (OAM) instruction over the channel being tested. The data used for I/O are the data patterns used in subsections 03 and 04. Every condition represents one pattern.

Subsection 02 (SUBSEC02)

Subsection 02 cannot be deselected. If subsections 03 or 04 are selected, the subsection 02 is selected. It tests the operation of the channel flag (channel flag number is equal to the receiving PP number). This channel flag is used in subsections 03 and 04 to control the receiving PP.

Subsection 03 (SUBSEC03)

Subsection 03 tests the data and Channel status during block I/O using the output block (OAM) instruction over the channel being tested. The size of the block is represented by the word count. Every condition is one block size (one word count).

Subsection 04 (SUBSEC04)

Subsection 04 tests the data and Channel status during block I/O using the output block (OAPM) instruction with conversion over the channel being tested. The size of the block is represented by the word count. Every condition is one block size (one word count).

CMA46 – Central Memory Access Test

The Central Memory Access Test (CMA46) checks the data path from PP memory to central memory (CM) and from CM to PP memory. It also checks the Relocation (R) register which, in conjunction with the A register, forms the CM address.

NOTE

When CMA46 is run in concurrent maintenance mode, the central memory buffer is forced to a 10,000 octal word buffer in the last Mbyte of central memory.

Section Descriptions

CMA46 is divided into ten sections.

Section	Tag	Description
00	SEC00	Tests the R register.
01	SEC01	Tests the (R+A), CM address.
02	CWDTST	Tests the read/write CM with 60-bit single CM word.
03	ADRTST	Tests the read/write addresses in CM with 60-bit single CM word.
04	CWMTST	Tests the read/write CM with 64-bit single CM word.
05	CLDTST	Tests the read/write block in CM with 60-bit CM word.
06	CLMTST	Tests the read/write block in CM with 64-bit CM word.
07	RSLTST	Tests the read and set lock.
08	RCLTST	Tests the read and clear lock.
09	MIXTST	Tests the read/write in mixed mode (60- and 64-bit CM word, single CM word and block).

Section 00 (SEC00)

The section has the following three subsections.

Subsection	Tag	Description
00	RRG00	Tests the lower 12 bits of the R register, with the upper 10 bits of R set to zero.
01	RRG01	Tests the upper 10 bits of the R register, with the lower 12 bits of R set to zero.
02	RRG02	Tests the full R register.

Every subsection has 1000_8 conditions and every condition is a different pattern.

Section 01 (SEC01)

The section has the following subsection.

Subsection	Tag	Description
00	RAATST	Tests the full A+R register.

This subsection has 1000_8 conditions and every condition is a different pattern.

Sections 02 through 09

Sections 02 through 09 have 16 subsections each. Subsections 00 through 07 use different starting addresses in CM and do not use the R register. Subsections 08 through 15 use randomly selected R register values.

Every subsection has 1000_8 conditions and every condition is a different pattern with a different starting address in CM.

Every condition is a different CM word count for the block read/write instruction.

MRA46 – Maintenance Register Access Test

The Maintenance Register Access Test (MRA46) checks the Maintenance register access from all PPs. It also checks the Maintenance Channel (MCH) hardware interlock and (MCH) priority circuits.

Section Descriptions

MRA46 has the following 26 sections.

Section	Tag	Description
00	SEC00	Tests the MCH interlock and PP priority.
01 through 25		Tests the MCH access from all PPs.

Section 00 (SEC00)

The section has the following two subsections.

Subsection	Tag	Description
00	S00B00	Tests that a PP from the lower barrel has priority over other PPs.
01	S00B01	Tests that only one PP can access the MCH at a time, using the MCH interlock.

Sections 01 through 25

Each section tests a MR access from the corresponding PP. Each section has the following five subsections.

Subsection	Tag	Description
00	FUNRES	Tests the response on function. Each condition is one function code.
01	ECHO	Tests the data to and from the MR using an echo function. Each condition is one pattern.
02	WRITE	Tests the Write function. The testing PP writes MR and the monitor PP reads and compares values.
03	READ	Tests the Read function. Each condition is one register in the MR.
04	SSUMF	Tests the Status-Summary function by reading data using the Status-Summary function and normal read operation.

MRT46 – Maintenance Register Test

The Maintenance Register Test (MRT46) checks all parity networks in the IOU using invert parity feature. All the force conditions that induce errors to other parts of the hardware are also checked.

The 16-digit expected and received messages are the expected and received contents of Fault Status 1 and Fault Status 2 registers.

NOTE

When MRT46 is run in concurrent maintenance mode, the central memory buffer is forced to a 10,000 octal word buffer in the last Mbyte of central memory.

Section Descriptions

Each PP is a section in test MRT46.

Subsection Descriptions

MRT46 is divided into the following 48 subsections:

Subsection	Tag	Description
00	SUB00	Tests for no reported errors.
01	SUB01	Tests invert parity on channel data to PPU.
02	SUB02	Tests invert parity from PPU to channel.
03	SUB03	Tests invert PPM to R parity.
04	SUB04	Tests invert parity at PPM data checker.
05	SUB05	Tests invert microcode parity.
06	SUB06	Tests invert PPM parity at parity generator.
07	SUB07	Tests invert parity on CM function code.
08	SUB08	Tests invert parity on Y register at parity generator.
09	SUB09	Tests invert parity on A register at parity generator.
10	SUB10	Tests invert parity on shift ROM at parity checker.
11	SUB11	Tests invert parity on Q register at parity generator.
12	SUB12	Tests invert parity on P register at parity generator.
13	SUB13	Tests invert parity on G register at parity generator.

Subsection	Tag	Description
14	SUB14	Tests invert R register to Y parity at checker.
15	SUB15	Tests invert PPM address parity at checker.
16	SUB16	Tests invert tag-out parity at generator.
17	SUB17	Tests invert data-in parity at generator.
18	SUB18	Tests invert CM address parity.
19	SUB19	Tests disable single error correction/double error detection (SECDED) code to PP memory.
20	SUB20	Tests invert channel input parity at checker.
21	SUB21	Tests for no error reported.
22	SUB22	Tests for no error reported.
23	SUB23	Tests for no error reported.
24	SUB24	Tests force CM request/resync error.
25	SUB25	Tests force mark line parity bit to zero.
26	SUB26	Tests force tag-out/in parity error.
27	SUB27	Tests force response code parity error.
28	SUB28	Tests invert function-out parity bit.
29	SUB29	Tests force address-out parity bit low.
30	SUB30	Tests force data-in parity bit low.
31	SUB31	Tests force data-out parity bit low.
32	SUB32	Tests invert channel parity from the two-port mux or the maintenance channel.
33	SUB33	Tests invert Maintenance register write parity.
34	SUB34	Tests invert nanocode parity.
35	SUB35	Tests invert read parity bit.
36	SUB36	Tests invert channel 15 data-bus parity at checker.
37	SUB37	Tests invert R/I read-data parity.
38	SUB38	Tests invert R/I write-data parity.
39	SUB39	Tests force lost clock error.
40	SUB40	Tests force CM request in the ADU.
41	SUB41	Tests force tag-in parity error in the ADU.

Subsection	Tag	Description
42	SUB42	Tests force response code parity error in the ADU.
43	SUB43	Tests block d5 full in the ADU.
44	SUB44	Tests force CMC busy at CMI.
45	SUB45	Tests invert OS bounds address parity at checker and the operation of the OS bounds register.
46	SUB46	Tests set inhibit PPU to CM request.
47	SUB47	Tests clear inhibit PPU to CM request.

NOTE

Any subsection that follows with an asterisk (*) next it indicates that it is not executed in concurrent maintenance mode.

Subsection 00 (SUB00)

This subsection tests that no errors are reported when a code of (00) is forced in the Test Mode register.

Subsection 01 (SUB01)

This subsection tests for invert parity on a channel to PPU data transfer.

Subsection 02 (SUB02)

This subsection tests for invert parity on a PPU to channel data transfer.

Subsection 03 (SUB03)

This subsection is executed to cause parity error for PPU to R register.

Subsection 04 (SUB04)

This subsection is executed to cause parity error at the PPU memory data checker.

Subsection 05 (SUB05)

This subsection causes parity errors in microcode and the G register.

Subsection 06 (SUB06)

This subsection tests the parity network for data sent to the PP memory.

Subsection 07 (SUB07)

This subsection tests the parity error on CM function.

Subsection 08 (SUB08)

This subsection tests the parity network of the Y register.

Subsection 09 (SUB09)

This subsection tests the A register parity error network.

Subsection 10 (SUB10)

This subsection shifts the data in A to cause shift control ROM parity error.

Subsection 11 (SUB11)

This subsection shifts the data for checking parity in the Q register.

Subsection 12 (SUB12)

This subsection cause parity errors in the P register.

Subsection 13 (SUB13)

This subsection causes a parity error in microcode and the G register.

Subsection 14 (SUB14)

This subsection is executed to cause parity error for R-to-Y transfer.

Subsection 15 (SUB15)

This subsection causes parity errors in the PPU memory address at the parity checker.

Subsection 16 (SUB16)

This subsection tests the data to CM with inverted tag-out parity.

Subsection 17 (SUB17)

This subsection tests data-in parity on CM transfers.

Subsection 18 (SUB18)

This subsection tests the parity error reporting when CM address parity bit is inverted.

Subsection 19 (SUB19)

This subsection tests the disable SECDED code generator to PP memory.

Subsection 20 (SUB20)

This subsection tests the invert parity of the channel input data at the input to the 12/16 conversion array.

Subsection 21 (SUB21)

This subsection tests that no errors are reported when a code of 25_8 is found in the Test Mode register.

Subsection 22 (SUB22)

This subsection tests that no errors are reported when a code of 26_8 is found in the Test Mode register.

Subsection 23 (SUB23)

This subsection tests that no errors are reported when a code of 27_8 is found in the Test Mode register.

Subsection 24 (SUB24)

This subsection tests for the CM request/resync error.

Subsection 25 (SUB25)*

This subsection tests the force mark line parity bit low.

Subsection 26 (SUB26)*

This subsection tests the force tag-out/in parity error.

Subsection 27 (SUB27)

This subsection tests the force response code parity error.

Subsection 28 (SUB28)*

This subsection tests the invert function-out parity bit.

Subsection 29 (SUB29)*

This subsection tests the force address out parity bits low.

Subsection 30 (SUB30)*

This subsection tests the parity error reporting when data-in parity is forced low on all groups.

Subsection 31 (SUB31)*

This subsection tests the force data-out parity bit low.

Subsection 32 (SUB32)

This subsection tests the invert channel parity on channel 17_8, or when the parity is inverted on channel 15_8 data bus.

Subsection 33 (SUB33)

This subsection tests the invert Maintenance register write parity on the data being written into the register.

Subsection 34 (SUB34)

This subsection tests the invert nanocode parity.

Subsection 35 (SUB35)

This subsection tests the invert read parity bit.

Subsection 36 (SUB36)

This subsection tests parity error reporting when the parity is inverted on channel 15 data bus.

Subsection 37 (SUB37)

This is the test for radial interface read.

Subsection 38 (SUB38)

This is the test for radial interface write.

Subsection 39 (SUB39)

This subsection tests the force lost clock error.

Subsection 40 (SUB40)

This subsection tests the force CM request at the ADU.

Subsection 41 (SUB41)

This subsection tests the force tag-in parity error at the ADU.

Subsection 42 (SUB42)

This subsection tests the force response code parity error at the ADU.

Subsection 43 (SUB43)

This subsection tests the force D5 full at the ADU.

Subsection 44 (SUB44)

This subsection tests the force CMC busy at the CMI.

Subsection 45 (SUB45)

This subsection tests invert OS bounds address parity at the checker and OS bounds violation when slave PP tries to write CM outside allowed region specified by OS bounds.

Subsection 46 (SUB46)

This subsection tests the set inhibit PP CM request.

Subsection 47 (SUB47)

This subsection tests the clear inhibit PP CM request.

TPM46 – Two-Port Multiplexer Test

The Two-Port Multiplexer Test (TPM46) is a standalone only test.

TPM46 tests the interface from and to the PP across channel 15_8 and port 1. The test requires user interaction and all faults are detected by the operator when the test stops with a display message error or a hung condition.

The test exercises only port 1 for functions that could interrupt normal PC console operation. The read and write clock sections are run from both ports and therefore test all ports.

The Clock sections test the special features available on the two-port mux.

The following message is displayed only at the beginning of the test when the set PARAM message is displayed:

```
SET DISPLAY TERMINAL

ON LINE,  PAGE MODE,  FULL DUP.

(LOC 141B)=PARITY SELECT

00=ODD,  01=EVEN,  02=NO PARITY

(LOC 142B)=STOP BITS

00=1 STOP BIT,  01=2  STOP BITS.

DEFAULT-EVEN PARITY/2 STOP BITS
```


Section Descriptions

TPM46 is divided into the following ten sections.

Section	Tag	Description
00	SEC00	Tests the Read Deadstart Port and Terminal function.
01	SEC01	Tests the function response and FIFO.
02	SEC02	Tests the DTR and Connect function.
03	SEC03	Tests the loopback with 5-, 6-, 7-, and 8-bit mode.
04	SEC04	Tests the Master Clear and Disconnect functions.

Section	Tag	Description
05	SEC05	Tests the data in overrun.
06	SEC06	Reads and displays the wall clock.
07	SEC07	Tests the wall clock by writing different test patterns.
08	SEC08	Tests the wall clock increment feature.
09	SEC09	Enters the time in the wall clock.

Section 00 (SEC00)

This section tests the Read Deadstart Port and Terminal function. The deadstart port and terminal are displayed on the console and operator input is required to continue the test.

Subsection	Tag	Description
00	SEC00.0	Reads and displays the deadstart port and terminal type.

Section 01 (SEC01)

The section has the following four subsections.

Subsection	Tag	Description
00	SEC01.4	Checks to see if the test initialization detected a response to the Connect function.
01	SEC01.6	Checks for the response on issued functions.
02	SEC01.8	Checks for FIFO operation.
03	SEC01.9	Checks if the DTR signal remains set if previously set by the DTR function after the FIFO has emptied.

Section 02 (SEC02)

The section has the following four subsections.

Subsection	Tag	Description
00	SEC02.1A	Tests the loopback feature with 8-bit mode.
01	SEC02.1	Tests the loopback feature with 7-bit mode.
02	SEC02.1	Tests the loopback feature with 6-bit mode.
03	SEC02.1	Tests the loopback feature with 5-bit mode.

Section 03 (SEC03)

The section has the following two subsections.

Subsection	Tag	Description
00	SEC03.0	Test the four character input buffer.
01	SEC03.0	Test the data in overrun bit. Verifies that the data in overrun bit will set if the data in the input buffer is overwritten.

Section 04 (SEC04)

The section has the following four subsections.

Subsection	Tag	Description
00	SEC04.2	Tests the clear Data Terminal Ready function (400) and checks for proper status on the port.
01	SEC04.6A	Tests the set Data Terminal Ready function and checks for proper status.
02	SEC04.8	Tests the Connect function (700X) by outputting on one port to fill the buffer and than connecting to the other port without interrupting the original output.
03	SEC04.b	Verifies that the two-port multiplexer will set the data terminal ready status bit when in Output mode.

Section 05 (SEC05)

The section has the following two subsections.

Subsection	Tag	Description
00	SEC05.1	Tests the Master Clear function (700). This subsection will display the message enter character. The character is to be entered by the operator that is required to fill the input buffer to test the Master Clear function.
01	SEC05.6	Tests the Disconnect function (6000).

Section 06 (SEC06)

The section has the following subsection.

Subsection	Tag	Description
00	SEC060	Reads and displays the calendar clock.

Section 07 (SEC07)

The section has the following subsection.

The subsection tests the Write and Read Wall Clock function by writing a pattern of data to the clock and then reading it back to verify. There are five fields defining the year (YY), month (MM), day (DD), hour (HH), and minute (MN).

```
:----------------------------------:
: 00 : YY : MM : DD : HH : MN : 00 :
:    :    :    :    :    :    :    :
```

Field	Condition	Patterns
*****	*********	********
YY	0	00
YY	1	01
.	.	.
.	.	.
YY	98	98
YY	99	99
MM	100	01
MM	101	02
.	.	.
.	.	.
MM	110	11
MM	111	12
DD	112	01
DD	113	02
.	.	.
.	.	.
DD	141	30
DD	142	31
HH	143	00
HH	144	01
.	.	.
.	.	.
HH	165	22
HH	166	23
MN	167	00
MN	168	01
.	.	.
.	.	.
MN	225	58
MN	226	59

In each condition, a pattern is written in a field; other fields contain 01. For example:

Condition 03: 00 03 01 01 01 01 00
Condition 98: 00 98 01 01 01 01 00
Condition 108: 00 01 09 01 01 01 00.

Section 08 (SEC08)

The section has the following three subsections.

Subsection	Tag	Description
00	SUBZER	This subsection checks propagation delay from minute-to-hour to day-to-month and to-year.

NOTE

To run subsections 1 and 2, clear quicklook mode bit.

01	SUBONE	This subsection checks the propagation of the carry (the increment of the clock). minute → hour, hour → day, day → month, month → year.
02	SUBTWO	This subsection tests the calendar clock to make sure that the last day is right for each month.

Section 09 (SEC09)

The section has the following subsection.

Subsection	Tag	Description
00	SEC090	This subsection can be used to set date and time in the wall clock from the keyboard. When the message ENTER DATE/TIME YYMMDDHHMN is displayed, enter data through the keyboard in the above form, where YY = year, MM = month, DD = day, HH = hour, MN = min. No separators are required. For example: 8506100812. If data is illegal an INVALID ENTRY message is displayed. If good data is entered, it is written to the clock, read back and displayed. Type ABS **CR** to exit or abort this subsection at the end of the test.

Isolation Diagnostics

The isolation diagnostic (FII46) analyzes errors reported by the detection tests and attempts to identify a group of logic paks that are likely to be causing the problem.

FII46 - Fault Isolation Program

The Fault Isolation Program (FII46) is a table driven analyzer that uses data recorded by the IOUCP6 Error Processing Routine (ERRCP) during IOU test execution.

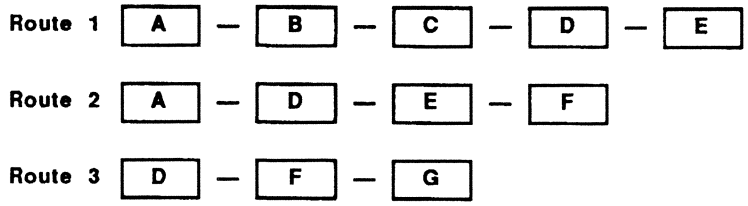
ERRCP records three classes of errors and assigns route numbers to them:

Error Class	Route Numbers
Unexpected parity errors	0
Invert parity errors	1 through 77_8
Data/control errors	100 through 177_8

FII46 analyzes the errors recorded by the detection tests using the route numbers as the base for isolation.

The route numbers for fault isolation are an index to a predefined string of modules.

Example:



FII46 sets error location (ERROR) to 20 and starts by processing route number 0. If an error is found in this route number, location (ERROR) is decremented by one for each pak that is isolated. If no errors are found for route number 0, FII46 processes route numbers 100 through 177_8. If no errors are found for route numbers 100 through 177_8, FII46 processes errors for route numbers 1 through 77_8.

When FII46 finds errors in the route numbers, it calls subroutines to find and identify the pak associated with the highest number of error routes. The subroutine assigns priority number one to paks with the highest count and then identifies lower priority paks. The maximum number of paks that can be called is 10.

Log Errors Routine

This routine is built into the IOUCP6 (IOU Control Program). It is used to log errors in the error buffer for fault isolation. It is called by IOUCP6 Error Processing Routine (ERRCP). The following classes of errors are logged:

- Unexpected parity error: When a parity error is detected, IOUCP6 generates a ROUTE=0 code and passes the contents of Fault Status registers 1 and 2 (FS1 and FS2) in (PEBUF) to the Error Log Routine.

- Invert parity error: When MRT42 detects an error, it generates a ROUTE=1 through 77_8 code and passes the logical difference between expected and received FS1 and FS2 in PEBUF to the Error Log Routine. The following parameters are also passed to the Error Log Routine.

  (EXPAT) Expected Pattern Buffer

  (RCPAT) Received Pattern Buffer

  (PPNUM) Failing PP Number

  (CHNUM) Failing CH Number

  (PEBUF) Parity Error Buffer

- Data/control error: When tests detect an error, they generate a ROUTE=100 through 177_8 code and passes the following parameters to the Error Log Routine.

  (EXPAT) Expected Pattern Buffer

  (RCPAT) Received Pattern Buffer

  (PPNUM) Failing PP Number

  (CHNUM) Failing CH Number

  (PEBUF) Parity Error Buffer

Figure 4-1 shows the Monitor PP map. The errors are recorded by Error Log Routine in the Error Log Buffer and processed by FII46.

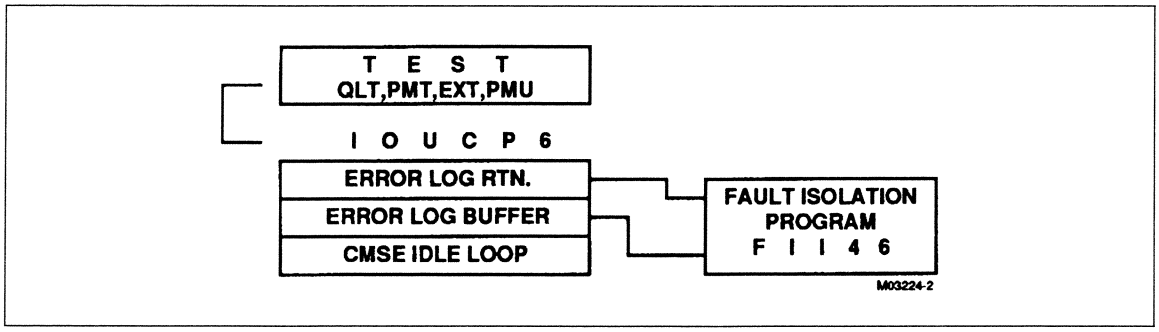


Figure 4-1. Monitor PP Map

Unexpected Parity Error Recording

The route number=0 is transmitted whenever unexpected parity errors occur. As shown in figure 4-2, the errors are logged by setting the RPEBUF equal to the accumulative logical difference of FS1 and FS2 from PEBUF, and PCOUNTER is incremented by one.

The size of the Recorded Parity Error Buffer (RPEBUF) is 20_8 locations.

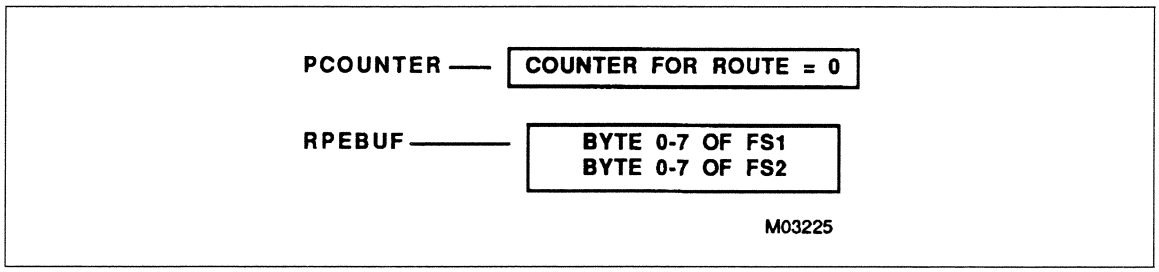


Figure 4-2. Parity Error Buffer

Invert Parity Error Recording

The routes numbered 1 through 77_8 are the invert parity error routes. When one of these errors is reported, the corresponding location (PCOUNTER+(ROUTE)) is incremented by one.

Also, a cumulative logical difference of FS1 and FS2 is recorded in location RPEBUF+20_8 to location RPEBUF+37_8.

Figure 4-3 shows what these locations represent.

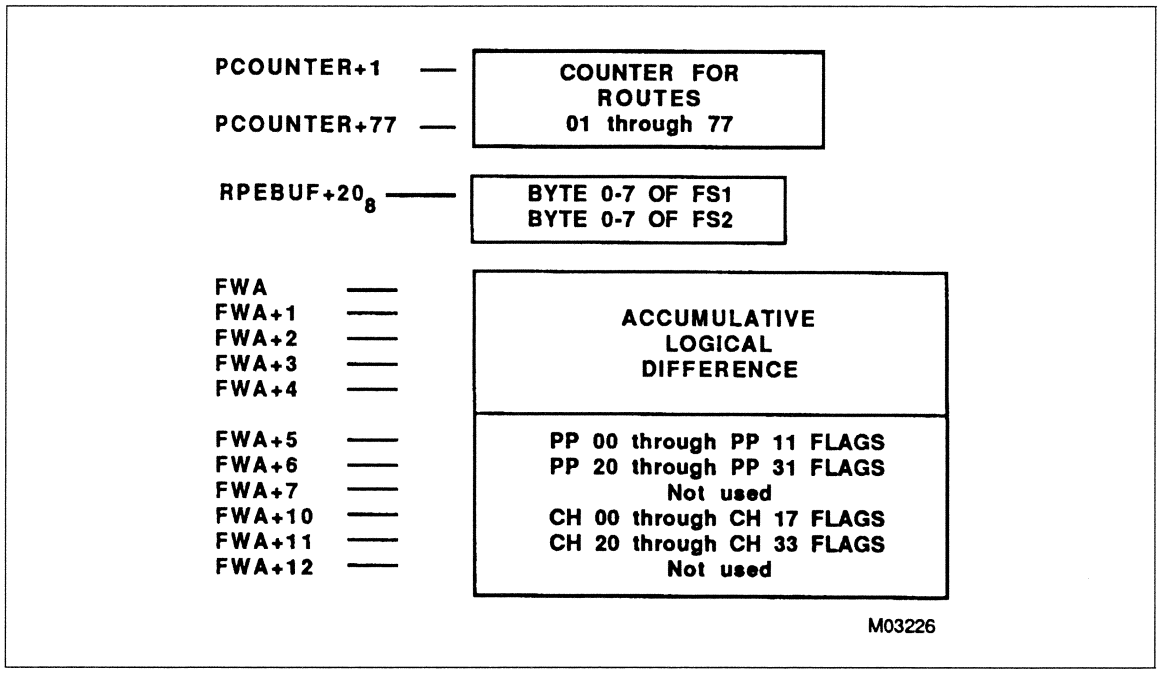


Figure 4-3. Invert Parity Error Buffer

Control and Data Error Recording

The routes numbered 100 through 177_8 represent the errors for data and control flow. The route numbers are used to generate the starting address for cumulative logical difference, PP, and channel flags when an error is detected.

The logical difference is calculated using expected (EXPAT) and received (RECPAT) patterns. The PP and channel position flags are calculated using (PPNUM) and (CHNUM).

The corresponding counter for failing route (PCOUNTER+(ROUTE)) is incremented by one.

Figure 4-4 shows the buffer for one failing route number.

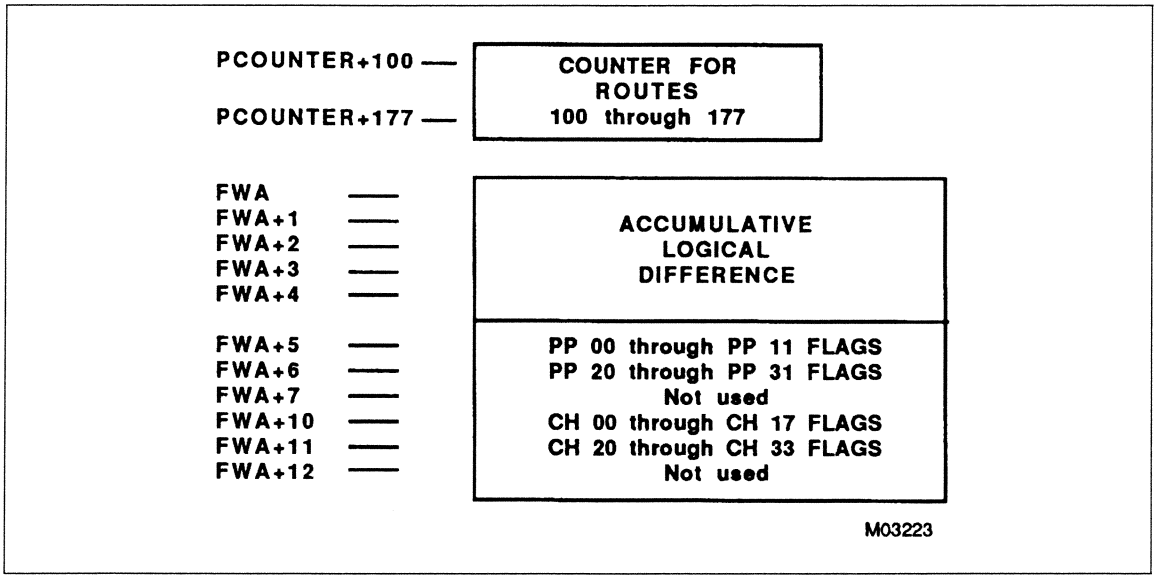


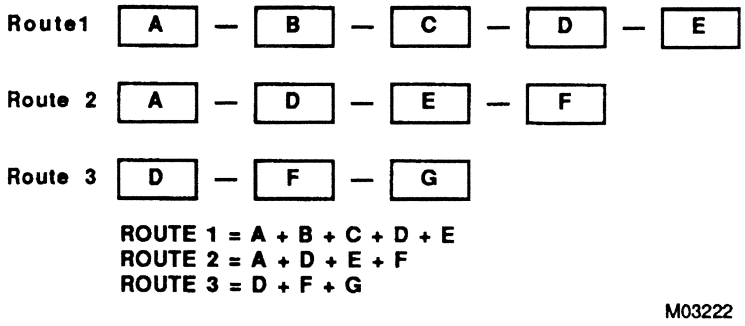
Figure 4-4. Control and Data Error

Fault Isolation Program

FII46 analyzes the errors recorded by the detection tests using route numbers as a base for isolation.

The route numbers for FII46 are considered as a index for the string of the involved modules.

Example:



FII46 sets location (ERROR) to 20_{10}, which is used as a flag when an error condition is detected.

The routine starts the processing of the unexpected parity error route (Route=0). If the unexpected parity error route is nonzero, location (ERROR) is decremented and the suspected failing paks are isolated. If location (ERROR) is equal to 20_{10}, route numbers 100 through 177_8 are processed. These route numbers represent the data and control flow errors. This processing is done by checking the corresponding location in the counter buffer for nonzero. When a location is found to be nonzero, location (ERROR) is decremented and the suspected failing paks are isolated.

If location (ERROR) is still equal to 20_{10}, the invert parity routes are processed. These routes represent the MRT46 detected errors. If there are no errors for these routes, FII46 calls CMSE to display the message NO ERROR RECORDED.

The FISORT routine is called if the unexpected parity error route or any of the data and control routes are nonzero. The FISORT route will find the pak with the highest number of calls. The counter for that pak has the highest count.

The FISOL routine is then called to set priority and identify the pak type and location. The calling table is passed to CMSE to display SUSPECTED FAILING MODULE/S (FISM1, Fault Isolation Message 1).

Parity Error Processing

This routine checks if location PCOUNTER is zero and, if it is, returns control to FICP with location (ERROR) unchanged. If PCOUNTER is nonzero, error location (ERROR) is decremented and the COPYPE routine is called to copy the recorded accumulative logical difference for this route in PEBUF and:

- Set (NUMBAR) number of failing barrels.
- Set (BARNUM) failing barrel number.

The parity error processor then increments the involved paks. If no error is found, it processes routes 100 through 177_8.

Data and Control Processing

The errors with route numbers 100_8 through 177_8 are processed in the following manner:

- Use COPYER routine to set the following:
  - Accumulative logical difference to received buffer.
  - Failing PP flags in PP parameter area.
  - Failing channel flags in channel parameter area.
  - Number of failing barrels (NUMBAR).
  - Failing physical barrel number (BARNUM).
- Call the processor for that route number to:
  - Determine if it is a data or control failure.
  - Find the number of failing bits and failing bit numbers.
  - Find the failing channel number and number of failing channels if it is channel failure.
  - Increment counters for the paks that caused the error.

Invert Parity Processing

This routine sets location (ERROR) to zero and checks if any location in the PCOUNTER buffer is nonzero and, if not, returns control to the Fault Isolation Control Program (FICP) with location (ERROR) unchanged. It sets location (ERROR) to nonzero if any failing route is found. If there is any error, it calls COPYPE routine to copy the recorded accumulative logical difference to PEBUF and:

- Set NUMBAR.
- Set BARNUM.

The routine then builds the display message for the suspected failing pak(s), which is displayed on the console.

Module Sorting

The Pak Sorting Routine (FISORT) is used to find the pak counter with the highest count and set Maximum Count Value (MAXCNT).

If the highest count is zero (processors were not able to determine the failing pak multiple uncorrelated error), the FISM2 (Fault Isolation Message 2) is passed to CMSE.

Fault Isolation

The Fault Isolation Routine (FISOL) is called to identify the pak type and pak location and to set the priority for the paks that have counters nonzero.

FISOL assigns priority number one to all the paks with the highest counters. It then decrements the maximum count and increments the priority number for the next set of paks with the same count as the current maximum count.

FISOL calls MODID for the pak location.

The maximum number of paks to be called is 10.

ISI4/ISI42/ISI44/ISI46 DMA-Enhanced ISI Channel Adapter 5

ISI4/ISI42/ISI44/ISI46 DMA-Enhanced ISI Channel Adapter 5

Section Descriptions

Section 0

This section tests the Intelligence Standard Interface (ISI) adapter access from the peripheral processor (PP). Channel signal response is verified by using the Master Clear Adapter function, channel input/output (I/O), and the Invert PP to channel parity facility in the Input/Output Unit (IOU) Maintenance register (MR).

Specific hardware tested includes:

- BAS Bus Interface, JX pak
- Channel Data register, JX pak
- Channel Data register parity checker, JX pak
- Error Status register, JX pak

Subsection 0

Tests the response on Master Clear Adapter function.

Method of Execution:

- Issue a Master Clear Adapter function to the ISI adapter.

The MAC Channel Status register is checked for errors.

```
< optional error data >    none.
```

Subsection 1

Tests the response on Read Error Status function.

Method of Execution:

1. Issue a Read Error Status function to the ISI adapter.

The Error status is read and checked for being equal to zero, and the CIO Fault Status 2 register is checked for errors.

```
< optional error data >    none.
```

Subsection 2

Tests the Channel Data register parity checker.

Method of Execution:

1. Enable the invert PP to channel parity Test mode facility in the IOU Maintenance register.
2. Output a data pattern on the channel and read.

The Error status and CIO Fault Status 2 registers are checked for errors. This subsection consists of four conditions with each condition being a different data pattern. The invert PP to channel parity Test mode must be enabled for each condition.

```
< optional error data >    PATTERN = pppp
```

Subsection 3

Tests the response on function parity error.

Method of Execution:

1. Enable the Invert PP to channel parity Test mode.
2. Function the ISI adapter.

The Error status and CIO Fault Status 2 registers are checked for errors.

```
< optional error data >    none.
```

Section 1

This section tests the ISI Adapter Control register using the Write Control register, Read Control register functions, and the Invert Control register parity Test mode facility.

Specific hardware tested includes:

- Input mux, JX pak
- Control register, JX pak
- Control register parity checker, JX pak
- Control register parity generator, JX pak
- MAC interface mux, JX pak
- MAC interface mux parity generator, JX pak

Subsection 0

Tests the interface between the Channel Data register and Control register.

Method of Execution:

1. Load data into the Control register using the Write Control register function.
2. Issue a Read Control register function and read data.

The Error Status register is checked for errors. This subsection consists of 18 conditions with each condition being a different data pattern.

```
< optional error data >    CONTROL REGISTER = cccc
```

Subsection 1

Tests the Control register parity generator.

Method of Execution:

1. Load data into the Control register.
2. Read the Control register.

The Error Status register is checked for errors. This subsection consists of 256 conditions with each condition being a different data pattern.

```
< optional error data >    CONTROL REGISTER = cccc
```

Subsection 2

Tests that the Master Clear Adapter function clears the Control register.

Method of Execution:

1. Load nonzero data into the Control register.
2. Issue a Master Clear Adapter function.

The register is then read and checked for being equal to zero.

```
< optional error data >     none.
```

Subsection 3

Tests the Control register parity checker.

Method of Execution:

- Load the Control register with the Invert Control register parity Test mode facility enabled.

The Error Status register and Channel Error Flag are checked for errors.

```
< optional error data >     none.
```

Subsection 4

Tests the interface between the Control register and the MAC Channel Status register.

Method of Execution:

1. Load data into the Control register.
2. Read the ISI adapter's Channel Status register.
3. Mask off the Control register bits.

The data and the Error Status register are checked for errors. This subsection consists of 256 conditions with each condition being a different data pattern.

```
< optional error data >     CONTROL REGISTER = cccc
```

Subsection 5

Tests force parity on the Function Decode Network.

Method of Execution:

- Enable the invert Prom Parity Test mode facility (force error decode 01) in the CYBER 170 adapter.

The Error Status register is checked for errors.

```
< optional error data >     none.
```

Section 2

Tests the ISI adapter's Flag Mask register utilizing the Write Flag Mask register, Read Flag Mask register functions, and the Invert Flag Mask register parity facility.

Specific hardware tested includes:

- Input mux, JX pak
- Flag Mask register, JX pak
- Flag Mask register parity checker, JX pak
- Flag Mask register parity generator, JX pak

Subsection 0

Tests the interface between the Channel Data register and Flag Mask register.

Method of Execution:

1. Load data into the Flag Mask register using the Write Flag Mask register function.
2. Issue a Read Flag Mask register function.
3. Read the Flag Mask register.

The Error Status register and data are checked for errors. This subsection consists of 18 conditions with each condition being a different data pattern.

```
< optional error data >    FLAG MASK = mmmm
```

Subsection 1

Tests the Flag Mask register parity generator.

Method of Execution:

1. Load data into the Flag Mask register.
2. Read the Flag Mask register.

The Error Status register and data are checked for errors. This subsection consists of 256 conditions with each condition being a different data pattern.

```
< optional error data >    FLAG MASK = mmmm
```

Subsection 2

Tests that the Master Clear Adapter function clears the Flag Mask register.

Method of Execution:

1. Load nonzero data into the Flag Mask register.
2. Issue a Master Clear Adapter function.

The register is read and data is checked to be zero.

```
< optional error data >     none.
```

Subsection 3

Tests the Flag Mask register parity checker.

Method of Execution:

1. Enable the Invert Flag Mask register parity.
2. Load data into the Flag Mask register.

The Error Status register and Channel Error Flag are checked for errors.

```
< optional error data >     none.
```

Section 3

This section tests the ISI adapter's operand generator using the Read Test Seed and Write Test Seed functions and the ISI adapter's Test mode facility in PP input mode.

Specific hardware tested includes:

- Operand generator, JZ pak
- Operand generator parity generator, JZ pak
- ISI 3-1 input mux, JZ pak
- ISI 3-1 input mux parity generator, JZ pak
- ISI input buffer, JZ pak
- ISI input buffer parity generator, JZ pak
- Input mux, JX pak

Subsection 0

Tests the operand generator and interface to the Channel Data register.

Method of Execution:

1. Load data into the operand generator using the Write Test Seed function.
2. Read the operand generator using the Read Test Seed function.

The Channel Error Flag and data are checked for errors. This subsection consists of 10 conditions with each condition being a different data pattern.

```
< optional error data >    TEST SEED ss
```

Subsection 1

Tests the operand generator to Channel Data register interface using the ISI input buffer.

Method of Execution:

1. Place the ISI adapter in Test mode.
2. Select the operand generator and load with zero.
3. Initiate a PP input transfer in Interlocked mode.
4. Read data.

The channel error flag and data are checked for errors. This subsection consists of 256 data reads from the ISI adapter channel.

```
< optional error data >
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Section 4

This section tests the interface between the Channel Data register and ISI channel receivers and transmitters using the ISI adapter Echo mode facility and invert/force parity.

Specific hardware tested includes:

- Output buffer data mux, JZ pak
- ISI output buffer, JZ pak
- ISI output buffer parity checker, JZ pak
- ISI output buffer parity generator, JZ pak
- Port A/B drivers, JZ pak
- Port A/B receivers, JZ pak
- ISI 3-1 input mux, JZ pak
- ISI 3-1 input mux parity checker, JZ pak
- ISI input buffer, JZ pak
- ISI input buffer parity checker, JZ pak

Subsection 0

Tests the PP to Port A ISI channel interface.

Method of Execution:

1. Place the ISI adapter in Echo mode.
2. Place a data pattern on the channel.
3. Switch the ISI adapter to Input mode and read data.

The channel error flag and data are checked for errors. This subsection consists of 18 conditions with each condition being a different data pattern.

```
< optional error data >    PATTERN = pppp
```

Subsection 1

Tests the PP to Port B ISI channel interface.

```
< optional error data >    PATTERN = pppp
```

Subsection 2

Tests the ISI output buffer parity checker.

Method of Execution:

1. Place the ISI adapter in Echo mode.
2. Enable invert ISI output buffer Byte 0 parity.
3. Place a data pattern on the channel.
4. Switch the ISI adapter to Input mode and read data.

The Error Status register, channel error flag, and data are checked for errors. This subsection consists of two conditions with each condition being a different data pattern. Invert ISI output buffer Byte 0 parity is used for each condition.

```
< optional error data >    PATTERN = pppp
```

Subsection 3

Tests the ISI output buffer parity checker.

Method of Execution:

- Same as subsection 2 except invert ISI output buffer Byte 1 parity is used for each condition.

```
< optional error data >    PATTERN = pppp
```

Subsection 4

Tests the ISI input mux parity checker.

Method of Execution:

1. Place the ISI adapter in Echo mode.
2. Enable port A.
3. Enable force zero Port A parity.
4. Place a data pattern on the channel.
5. Switch the ISI adapter to Input mode and read data.

The Error Status register, channel error flag, and data are checked for errors. This subsection consists of two conditions with each condition being a different data pattern. Force zero Port A parity is used for each condition.

```
< optional error data >    PATTERN = pppp
```

Subsection 5

Tests port select.

Method of Execution:

1. Place the ISI adapter in Echo mode.
2. Enable Port A.
3. Enable force zero Port B parity.
4. Place a data pattern on the channel.
5. Switch the ISI adapter to Input mode and read data.

The Error Status register, channel error flag, and data are checked for errors.

```
< optional error data >    PATTERN = pppp
```

Subsection 6

Tests the ISI input buffer parity checker.

Method of Execution:

1. Place the ISI adapter in Echo mode.
2. Enable Port A.
3. Enable Force Zero input buffer parity.
4. Place a data pattern on the channel.
5. Switch the ISI adapter to Input mode and read data.

The Error Status register, Channel Error Flag, and data are checked for errors. This subsection consists of two conditions with each condition being a different data pattern. Force Zero Port A parity is used for each condition.

```
< optional error data >    PATTERN = pppp
```

Subsection 7

Tests the ISI output buffer parity generator.

Method of Execution:

1. Place the ISI adapter in Echo mode.
2. Output a data word on the channel.
3. Switch the ISI adapter to Input mode and read data.

The Channel Error Flag and data are checked for errors. This subsection consists of 256 conditions with each condition being a different data pattern.

```
< optional error data >    PATTERN = pppp
```

Subsection 8

Tests the ISI input buffer.

Method of Execution:

1. Place the ISI adapter in Echo mode.
2. Output a block of 16 data words on the channel.
3. Switch the ISI adapter to Input mode and read data.

The Channel Error Flag and data are checked for errors. This subsection consists of 16 conditions with each condition being a different block of 16 data words.

```
< optional error data >    none.
```

Subsection 9

Tests the ISI input buffer Overflow Response.

Method of Execution:

1. Place the ISI adapter in Echo mode.
2. Enable Force Inhibit Outstanding Request counter.
3. Ouput a block of 16 data words on the channel.
4. Attempt to output another data word.

The Error Status register and Channel Error Flag are checked for errors.

```
< optional error data >    none
```

Section 5

This section tests the ISI Adapter Compare Unit using the Test mode facility in PP Output mode.

Specific hardware tested includes:

- ISI output buffer, JZ pak
- Operand generator, JZ pak
- Compare unit, JZ pak

Subsection 0

Tests the Compare unit.

Method of Execution:

1. Place the ISI adapter in Test mode.
2. Select the operand generator in inhibit increment mode.
3. Load the first operand into the operand generator.
4. Initiate a PP output transfer in interlocked mode.
5. Place the second operand on the channel.

The TM error bit in the Error Status register and the Channel Error Flag are checked for errors. This subsection consists of 61 conditions with each condition being a different pair of operands.

```
< optional error data >
TEST SEED ss
DATA PTRN pppp
XFER BLOCK  cccc bbbb tttt tttt
```

Section 6

This section is the ISI channel backup test. The Test mode facility in PP output mode is used.

Specific hardware tested includes:

- ISI output buffer, JZ pak

Subsection 0

Tests the ISI output buffer.

Method of Execution:

1. Place the ISI adapter in Test mode with Inhibit Sync Out enabled.
2. Select the operand generator in inhibit increment mode.
3. Load a data pattern into the operand generator.
4. Initiate a PP output transfer in interlocked mode.
5. Output the data pattern four times.
6. Disable Inhibit Sync Out.

The Channel Error Flag is checked for errors. This subsection consists of 10 conditions with each condition being a different data pattern.

```
< optional error data >
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Section 7

This section tests ISI channel timeout.

Specific hardware tested includes:

- ISI timeout counter, JZ pak

Subsection 0

Test for ISI timeout.

Method of Execution:

- Set the Broadcast Master Reset bit in the Bus Unit Select word during a Bus Unit Select operation.

An ISI timeout occurs after one second. The Channel Error Flag and the Error Status register are checked for errors.

```
< optional error data >     none
```

Subsection 1

Test inhibit ISI timeout.

Method of Execution:

- Same as subsection 0, except the disable ISI timeout bit is set in the Control register and no errors are expected.

```
< optional error data >     none
```

Section 8

This section tests ISI channel Idle mode.

Specific hardware tested includes:

- Flag Mask register Enable Channel Flag Network, JX pak

Subsection 0

Tests the Request ISI Idle Status function.

Method of Execution:

1. Issue a Request ISI Idle Status function.
2. Read the Idle Bus status word.

The Channel Error Flag and the Error Status register are checked for errors.

```
< optional error data >    none.
```

Subsection 1

Test the Enable Channel Flag Network.

Method of Execution:

1. Place the ISI adapter in Echo mode.
2. Enable Idle Test mode.
3. Load the Flag Mask register with a data pattern.
4. Output the data pattern on the channel.
5. Switch the ISI adapter to Input mode and read data.

The Channel Flag, Channel Error Flag, and data are checked for errors. This subsection consists of eight conditions with each condition being a different data pattern.

```
< optional error data >    FLAG MASK AND DATA ffff
```

Subsection 2

Test transfer complete doesn't set the Channel Flag in PP mode.

Method of Execution:

1. Place the ISI adapter in Echo mode.
2. Set the transfer complete bit in the Flag Mask register.
3. Output a data pattern on the channel and input.

The Channel Flag and Error Status register are checked for errors.

```
< optional error data >     none
```

Subsection 3

Test response on Request Bit Significant Response.

Method of Execution:

1. Set command sequence.
2. Send a Zero Bus Unit Select word.
3. Send the Force Sync Out function.
4. Read the Bit Significant Response.

The Bit Significant Response is compared with all ones.

```
< optional error data >     none
```

Section 9

This section is the PP transfer test.

Subsection 0

Tests PP transfers in interlocked mode.

Method of Execution:

1. Place the ISI adapter in Test mode.
2. Select and initialize the operand generator with a random data pattern.
3. Initiate a PP input transfer in interlocked mode.
4. Read 200 data words from the channel.
5. Check the Channel Error Flag for errors.
6. Reset the operand generator to the random data pattern.

Data read is used as output data after initiating a PP output transfer in interlocked mode. The Channel Error Flag is checked for errors. This subsection consists of 10 conditions with each condition being a different random data pattern.

```
< optional error data >
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 1

Tests PP transfers in noninterlocked mode.

Method of Execution:

- Same as subsection 0, except noninterlocked mode is used.

```
< optional error data >
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Section 10

This section tests the DMA Transfer registers, using the Clear T register, Read T register, and Write T register functions and force parity.

Specific hardware tested includes:

- T registers driver, JY pak
- T register, JY pak
- T register, JY pak
- T register 3-1 Output mux, JY pak
- T register Byte Count parity checker, JY pak
- T register parity checker, JY pak
- Flag Mask register Enable Channel Flag Network, JX pak
- Operational Status register, JX pak

Subsection 0

Tests the Write T register function.

Method of Execution:

1. Issue a Write T register function.
2. Output three words.
3. Attempt to output a fourth word.

The Channel signal response and Error Status register are checked for errors.

```
< optional error data >     none
```

Subsection 1

Tests the Clear T register function.

Method of Execution:

1. Load the T register with a data pattern.
2. Issue a Clear T register function.
3. Read the contents of the T register and verify it to be zero.

The Error Status register is checked for errors.

```
< optional error data >
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 2

Tests the interface between the T registers and the Channel Data register.

Method of Execution:

1. Load data into the T register using the Write T register function.
2. Read the T register using the Read T register function.

The Error Status register and data are checked for errors.

```
< optional error data >
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 3

Tests the T register parity checkers.

Method of Execution:

1. Enable Force T register parity.
2. Master clear the ISI adapter.
3. Load a data pattern into the T register.

The Error Status register and Channel Error Flag are checked for errors. This subsection consists of seven conditions with each condition being a different data pattern.

```
< optional error data >
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 4

Test the T register parity generators.

Method of Execution:

1. Load the T register with a data pattern.
2. Read the contents of the T register is read.

The Channel Error Flag and data are checked for errors. This subsection consists of seven conditions with each condition being a different data pattern.

```
< optional error data >
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 5

Test the T register Empty status.

Method of Execution:

1. Set the T register Empty bit in the Flag Mask register.
2. Issue a Clear T register function and the Channel Flag is cleared.
3. Load the T register with a data pattern.

The Channel Flag and Operational Status register are checked for errors.

1. Clear the Channel Flag.
2. Load the T register with the same data pattern.

The Channel Flag and Operational Status register are checked for errors.

```
< optional error data >
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 6

Tests response on T register parity error.

Method of Execution:

1. Enable Invert T register parity.
2. Load a data pattern into the T register.

The Error Status register is checked for errors.

```
< optional error data >
XFER BLOCK  cccc bbbb tttt tttt
```

Section 11

This section tests the central memory (CM) to ISI Transfer register interface using the Test mode facility in DMA read mode and invert parity.

Specific hardware tested includes:

- Output buffer data mux, JZ pak
- Disassembly buffer, JY pak
- Disassembly buffer Output register, JY pak
- Disassembly buffer parity checker, JY pak
- T register, JY pak

Subsection 0

Tests ISI Transfer-In-Progress.

Method of Execution:

1. Place the ISI adapter in Test mode with ISI timeout disabled, Sync Out inhibited, and Test mode increment inhibited.
2. Select the operand generator and initialize with a PP unique data pattern.
3. Initialize the CM buffer.
4. Load the T register with the CM address and byte count for a one-word transfer.
5. Initiate a DMA read transfer in noninterlocked mode.
6. Check the Operational Status register to ensure that Transfer-In-Progress is set.
7. Enable Sync Out and ISI timeout to allow the transfer to complete.

```
< optional error data >
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 1

Tests Transfer Complete setting of Channel Flag.

Method of Execution:

1. Place the ISI adapter in Test mode.
2. Select the operand generator and initialize with a data pattern.
3. Initialize the CM buffer.
4. Load the T register with the CM address and byte count for a one-word transfer.
5. Initiate a DMA read transfer in interlocked mode.

The Channel Flag and Error status are checked for errors.

```
< optional error data >
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 2

Tests the Central Memory to ISI Transfer register interface.

Method of Execution:

1. Place the ISI adapter in Test mode.
2. Select the operand generator and initialize with a data pattern.
3. Initialize the CM buffer.
4. Load the T register with the CM address and byte count for a one-word transfer.
5. Initiate a DMA read transfer in interlocked mode.

The NIO Fault Status 1 and Error Status registers are checked for errors. This subsection consists of 13 conditions with each condition being a different data pattern. The real memory address and byte count remain unchanged for each condition.

```
< optional error data >
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 3

Tests the disassembly buffer ranks.

Method of Execution:

1. Place the ISI adapter in Test mode with Inhibit Sync Out enabled.
2. Select the operand generator and initialize with a data pattern.
3. Load the T register.
4. Initiate a DMA read transfer in Interlocked mode.

When the transfer stops, Inhibit Sync Out is disabled allowing the transfer to complete. The NIO Fault Status 1 and Error Status registers are checked for errors. This subsection consists of 13 conditions with each condition being a different data pattern. The real memory address and byte count remain unchanged for each condition.

```
< optional error data >
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 4

Tests the disassembly buffer parity generators.

Method of Execution:

1. Place the ISI adapter in Test mode.
2. Initialize the CM buffer with a data pattern.
3. Load the T register with the CM address and byte count for a one-word transfer.
4. Initiate a DMA read transfer in Interlocked mode.

The NIO Fault Status 1 and Error Status registers are checked for errors. This subsection consists of 72 conditions with each condition being a different data pattern. The real memory address and byte count remain unchanged for each condition.

```
< optional error data >
TEST SEED ss
DATA PTRN pppp
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 5

Tests the disassembly buffer lower byte parity checker.

Method of Execution:

- Place the ISI adapter in Test mode with invert disassembly buffer lower byte parity enabled.

The NIO Fault Status 1, CIO Fault Status 2, and Error Status registers are checked for errors.

```
< optional error data >     none
```

Subsection 6

Tests the disassembly buffer upper byte parity checker.

Method of Execution:

- Same as subsection 5 except invert disassembly buffer upper byte parity is used.

```
< optional error data >     none.
```

Section 12

This section tests the ISI Transfer register to CM interface utilizing the Test mode facility in DMA write mode and invert parity.

Specific hardware tested includes:

- Input data fan-in, JZ pak
- Assembly buffer, JY pak
- DMA input data buffer, JZ pak
- Assembly output buffer, JY pak

Subsection 0

Tests ADU Transfer-In-Progress. This subsection is bypassed if PARAM15 is set.

Method of Execution:

1. Place the ISI adapter in Test mode with Test mode increment inhibited.
2. Select the operand generator and initialize with a PP unique data pattern.
3. Block CM requests at the CM interface.
4. Load the T register with the CM address and byte count for a one-word transfer.
5. Initiate a DMA write transfer in non-interlocked mode.
6. Check the Operational Status register to insure that Transfer-In-Progress is set.

Central memory requests are released at the CM interface to allow the transfer to complete.

```
< optional error data >
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 1

Tests the ISI Transfer register to CM interface.

Method of Execution:

1. Place the ISI adapter in Test mode.
2. Select the operand generator and load with a data pattern.
3. Load the T register with the CM address and byte count for a one-word transfer.
4. Initiate a DMA write transfer in interlocked mode and data in CM is checked.

The NIO Fault Status 1 and Error Status registers are checked for errors. This subsection consists of 13 conditions with each condition being a different data pattern. The real memory address and byte count remain unchanged for each condition.

```
< optional error data >
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 2

Tests the assembly buffer ranks. This subsection is bypassed if PARAM15 is set.

Method of Execution:

1. Place the ISI adapter in Test mode.
2. Block CM requests at CM interface.
3. Initiate a DMA write transfer in noninterlocked mode.

When the transfer stops, CM requests are allowed to CM allowing the transfer to complete and data in CM is checked for errors. The NIO Fault Status 1 and Error Status registers are checked for errors. This subsection consists of 13 conditions with each condition being a different data pattern.

```
< optional error data >
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 3

Tests the assembly buffer parity generators.

Method of Execution:

1. Place the ISI adapter in Test mode.
2. Load the T register with the CM address and byte count for a one-word transfer.
3. Select the operand generator and initialize with a data pattern.
4. Initiate a DMA write transfer in interlocked mode and data in CM is checked for errors.

The NIO Fault Status 1 and Error Status registers are checked for errors. This subsection consists of four conditions with each condition being a different data pattern.

```
< optional error data >
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 4

Tests the assembly buffer lower byte parity checker.

Method of Execution:

1. Place the ISI adapter in Test mode with force assembly buffer lower byte parity enabled.
2. Load the T register with the CM address and byte count for a one-word transfer.
3. Select the operand generator and initialize with a data pattern.
4. Initiate a DMA write transfer in interlocked mode and data is checked for errors.

The NIO Fault Status 1 and Error Status registers are checked for errors. This subsection consists of three conditions with each condition being a different data pattern.

```
< optional error data >
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 5

Tests the assembly buffer upper byte parity checker.

Method of Execution:

- Same as subsection 4 except force assembly buffer upper byte parity is used.

```
< optional error data >
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Section 13

This section tests the DMA Transfer register address incrementer using the Test mode facility in DMA Read mode.

Specific hardware tested includes:

- T register real memory address incrementer, JY pak
- T register parity prediction network, JY pak

Subsection 0

Tests the DMA Transfer register address incrementer.

Method of Execution:

1. Place the ISI adapter in Test mode.
2. Select the operand generator and initialize with a data pattern.
3. Load the T register with a CM address data pattern.
4. Set the byte count for a one-word transfer.
5. Initialize the CM address that will be read if the address is within the defined CM buffer boundaries.
6. Initiate a DMA read transfer in interlocked mode.

The NIO Fault Status 1, Error Status registers, and T register are read and checked for errors. This subsection consists of 56 conditions with each condition being a different CM address pattern.

```
< optional error data >
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 1

Tests force parity prediction error.

Method of Execution:

1. Place the ISI adapter in Test mode with invert parity prediction enabled.
2. Load the T register with a CM address and byte count for a one-word transfer.
3. Initiate a DMA write transfer in interlocked mode.

The NIO Fault Status 1, Error Status registers, and data in CM are checked for errors.

```
< optional error data >
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Section 14

This section tests the DMA Transfer register byte counter decrementer utilizing the Test mode facility in DMA read mode.

Specific hardware tested includes:

- T register byte count decrementer, JY pak

Subsection 0

Tests the DMA Transfer register byte counter decrementer.

Method of Execution:

1. Place the ISI adapter in Test mode.
2. Select the operand generator and initialize with a PP unique pattern.
3. Load the T register with the maximum byte count available.
4. Initialize the CM buffer.
5. Initiate a DMA read transfer in interlocked mode and data in CM is checked for errors.

The NIO Fault Status 1 and Error Status registers are checked for errors.

```
< optional error data >
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Section 15

This section is the DMA transfer test.

Subsection 0

Tests incomplete word DMA read transfers.

Method of Execution:

- Initiate a DMA read transfer in interlocked mode with an incomplete word byte count.

The NIO Fault Status 1 and Error Status registers are checked for errors.

```
< optional error data >
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 1

Tests incomplete word DMA write transfers.

Method of Execution:

1. Initiate a DMA write transfer in interlocked mode with an incomplete word byte count.
2. Check data in CM for errors.

The NIO Fault Status 1 and Error Status registers are checked for errors.

```
< optional error data >
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 2

Tests DMA transfers in Interlocked mode.

Method of Execution:

1. Place the ISI adapter in Test mode.
2. Select the operand generator and initialize with a random data pattern.
3. Load the T register with the starting CM address and byte count.
4. Initiate a DMA write transfer in Interlocked mode.
5. Reload the T registers and operand generator with the same values.
6. Initiate a DMA read transfer in Interlocked mode.

The NIO Fault Status 1 and Error Status registers are checked for errors. This subsection consists of 10 conditions with each condition being a different data pattern.

```
< optional error data >
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 3

Tests DMA transfers in Noninterlocked mode.

Method of Execution:

- Same as subsection 2, except transfers are in Noninterlocked mode.

```
< optional error data >
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Section 16

This section is the DMA transfer error test. Errors are forced in this section by using the DMA adapter's hardware functions.

NOTE

Subsections 0, 3, 4, 5, and 6 are not executed when in concurrent maintenance mode.

Subsection 0

Tests response on CM response code parity error.

Method of Execution:

1. Enable invert CM response code parity.
2. Initiate a DMA write transfer in Interlocked mode.

The Error Status register, NIO Fault Status 1 register, CIO Fault Status 2 register, and data in CM are checked for errors.

```
< optional error data >
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 1

Tests response on invalid response code.

Method of Execution:

1. Enable force invalid response code.
2. Initiate a DMA write transfer in Interlocked mode.

The Error Status register, NIO Fault Status 1 register, CIO Fault Status 2 register, and data in CM are checked for errors.

```
< optional error data >
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
register, and data in CM are checked.
```

Subsection 2

Tests response on force byte count zero.

Method of Execution:

1. Enable Force Byte Count Zero function.
2. Initialize the CM buffer.
3. Initiate a DMA read transfer in Interlocked mode.

The Error Status register, NIO Fault Status 1 register, CIO Fault Status 2 register, and data in CM are checked for errors.

```
< optional error data >
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 3

Tests false start transfer, byte count zero.

Method of Execution:

1. Initialize the CM buffer with the PP unique data pattern.
2. Initiate a DMA write transfer in interlocked mode using a byte count of zero.

The Error Status register, NIO Fault Status 1 register, CIO Fault Status 2 register, and data in CM are checked for errors.

```
< optional error data >
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Section 17

This section is the DMA transfer error test. Errors are forced in this section by using the Maintenance register hardware functions. This section is bypassed if PARAM15 is set.

Subsection 0

Tests response on uncorrected write error.

Method of Execution:

1. Enable Force Data-Out Parity function at Central Memory Interface (CMI).
2. Initiate a DMA write transfer.

The Error Status register, NIO Fault Status 1 register, and CIO Fault Status 2 register are checked for errors.

```
< optional error data >
MR TEST MODE =mm
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 1

Tests response on uncorrected read error.

Method of Execution:

1. Enable Force Data-In Parity function at CMI.
2. Initiate a DMA read transfer.

The Error Status register, CIO Channel Status register, NIO Fault Status 1 register, and CIO Fault Status 2 register are checked for errors.

```
< optional error data >
MR TEST MODE =mm
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 2

Tests response on CMI response code parity error.

Method of Execution:

1. Enable Force Response Code Parity Error function at CMI.
2. Initiate a DMA write transfer.

The Error Status register, NIO Fault Status 1 register, and CIO Fault Status 2 register are checked for errors.

```
< optional error data >
MR TEST MODE =mm
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 3

Tests response on CMI reject.

Method of Execution:

1. Enable Invert Function Out Parity function at CMI.
2. Initiate a DMA read transfer.

The Error Status register, CIO Channel Status register, NIO Fault Status 1 register, and CIO Fault Status 2 register are checked for errors.

```
< optional error data >
MR TEST MODE =mm
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```


Subsection 4

Tests response on address out parity error.

Method of Execution:

1. Enable address out Parity function at CMI.
2. Initiate a DMA read transfer.

The Error Status register, CIO Channel Status register, NIO Fault Status 1 register, and CIO Fault Status 2 register are checked for errors. This subsection is repeated for an address bit in each parity byte.

```
< optional error data >
MR TEST MODE =mm
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 5

Tests response on tag-in/out errors.

Method of Execution:

1. Enable Force Tag-In/Out parity error function at CMI.
2. Initiate a DMA transfer.

The Error Status register, NIO Fault Status 1 register, and CIO Fault Status 2 register are checked for errors. This subsection is repeated for transfers in both directions.

```
< optional error data >
MR TEST MODE =mm
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Subsection 6

Tests response on mark parity bit errors.

Method of Execution:

1. Enable Force Mark Parity Bit to Zero function at CMI.
2. Initiate a DMA transfer.

The Error Status register, NIO Fault Status 1 register, and CIO Fault Status 2 register are checked for errors. This subsection is repeated for transfers in both directions.

```
< optional error data >
MR TEST MODE =mm
TEST SEED ss
XFER BLOCK  cccc bbbb tttt tttt
```

Section 18

This section is the stacked transfer test. A minimum of six transfers are sequentially sent to the ISI adapter.

Subsection 0

Method of Execution:

1. Place the ISI adapter in Test mode.
2. Select the operand generator and initialize with a random data pattern. The byte counts are generated randomly. The starting real memory addresses are computed from the byte counts.
3. Initiate DMA write transfers until the byte count/address pairs have expired.
4. Reload the operand generator with the random data pattern.
5. Initiate DMA read transfers with the same byte count/address pairs to check the data.

The NIO Fault Status 1 and Error Status registers are checked for errors. This sequence is repeated for all combinations of interlocked and noninterlocked transfers.

```
< optional error data >
TEST SEED ss
XFER NO.  nn
NXT cccc bbbb tttt tttt
CUR cccc bbbb tttt tttt
```

CCA4/CCA42/CCA44/CCA46 DMA-Enhanced CYBER 170 Channel Adapter 6

CCA4/CCA42/CCA44/CCA46 DMA-Enhanced CYBER 170 Channel Adapter 6

Section Descriptions

Section 0

This section tests the CYBER 170 Channel adapter access from the peripheral processor (PP). Channel signal response is verified by using the Master Clear Adapter function, channel input/output (I/O) and the invert PP to Channel Parity Test mode facility in the input/output unit (IOU) Maintenance register.

Specific hardware tested includes:

- BAS bus interface, KX pak
- Channel Data register, KX pak
- Channel Data register parity checker, KX pak
- Error Status register, KX pak

Subsection 0

Tests the response on Master Clear Adapter function.

Method of Execution:

- Issue a Master Clear Adapter function to the CYBER 170 adapter.

The MAC Channel Status register is checked for errors.

```
< optional error data >    none.
```

Subsection 1

Tests the response on Read Error Status function.

Method of Execution:

1. Issue a Master Clear Adapter function to the CYBER 170 adapter.
2. Issue a Read Error Status function.

The Error Status is read and checked for being equal to zero, and the CIO Fault Status 2 register is checked for errors.

```
< optional error data >    none.
```

Subsection 2

Tests the Channel Data register parity checker in the CYBER 170 adapter.

Method of Execution:

1. Enable the Invert PP To Channel Parity Test mode facility in the IOU Maintenance register.
2. Output a data pattern on the channel and read.

The Error Status and CIO Fault Status 2 registers are checked for errors. This subsection consists of four conditions with each condition being a different data pattern. The invert PP to Channel Parity Test mode must be enabled for each condition.

```
< optional error data >    PATTERN = pppp
```

Subsection 3

Tests the response on function parity error in the CYBER 170 adapter.

Method of Execution:

1. Enable the Invert PP To Channel Parity Test mode facility in the IOU Maintenance register.
2. Issue a Write Control register function to the CYBER 170 adapter.

The Error Status and CIO Fault Status 2 registers are checked for errors.

```
< optional error data >    none.
```

Subsection 4

Tests the operation of the 170 Mode Channel Error Flag and Adapter Error Status register in the CYBER 170 adapter.

Method of Execution:

1. Enable the Invert PP To Channel Parity Test mode facility in the IOU Maintenance register.
2. Issue a Write Control register function to the CYBER 170 adapter.

The Channel Error Flag is checked for being set and the Error Status register is checked for being equal to zero.

```
< optional error data >    none.
```

Section 1

This section tests the CYBER 170 adapter Control register using the Write Control register, Read Control register functions, and the Invert Control register parity Test mode facility in the CYBER 170 adapter.

Specific hardware tested includes:

- Input mux, KX pak
- Control register, KX pak
- Control register parity checker, KX pak
- Control register parity generator, KX pak
- MAC interface mux, KX pak
- MAC interface mux parity generator, KX pak

Subsection 0

Tests the interface between the Channel Data register and the Control register in the CYBER 170 adapter.

Method of Execution:

1. Load data into the Control register using the Write Control register function.
2. Issue a Read Control register function and read data.

The Error Status register is checked for errors. This subsection consists of 18 conditions with each condition being a different data pattern.

```
< optional error data >    CONTROL REGISTER=cccc
```

Subsection 1

Tests the Control register parity generator in the CYBER 170 adapter.

Method of Execution:

1. Load data into the Control register.
2. Read the Control register.

The Error Status register is checked for errors. This subsection consists of 256 conditions with each condition being a different data pattern.

```
< optional error data >    CONTROL REGISTER=cccc
```

Subsection 2

Tests that the Master Clear Adapter function clears the Control register in the CYBER 170 adapter.

Method of Execution:

1. Load nonzero data into the Control register.
2. Issue a Master Clear Adapter function.

The Control register is then read and checked for being equal to zero.

```
< optional error data >     none.
```

Subsection 3

Tests the Control register parity checker in the CYBER 170 adapter.

Method of Execution:

- Load the Control register in the CYBER 170 adapter with the Invert Control Register Parity Test mode facility (force error code 06) enabled.

The Error Status register is checked for errors.

```
< optional error data >     none.
```

Subsection 4

Tests the Disable 170 Mode Error Flag bit in the Control register.

Method of Execution:

- Load the Control register in the CYBER 170 adapter with the Invert Control Register Parity Test mode facility (force error code 06) enabled and the Disable CYBER 170 Error Flag bit set.

The Channel Error Flag is checked for being set and the Error Status register is checked for errors.

```
< optional error data >     none.
```

Subsection 5

Tests the interface between the Control register and the MAC Channel Status register in the CYBER 170 adapter.

Method of Execution:

1. Load data into the Control register.
2. Read the CYBER 170 adapter Channel Status register.
3. Mask off the Control register bits.

The data and the Error Status register are checked for errors. This subsection consists of 256 conditions with each condition being a different data pattern.

```
< optional error data >    CONTROL REGISTER=cccc
```

Subsection 6

Tests Force Parity Decode Network in the CYBER 170 adapter.

Method of Execution:

- Enable the Invert Prom Parity Test mode facility (force error code 01) in the CYBER 170 adapter.

The Error Status register is checked for errors in the CYBER 170 adapter.

```
< optional error data >    none.
```

Subsection 7

Tests the PP input data parity using the Control register in the CYBER 170 adapter.

Method of Execution:

1. Enable the Invert PP Input Data Parity Test mode facility (force error code 02) in the CYBER 170 adapter.
2. Read the Control register.

The Error Status and CIO Fault Status 2 registers are checked for errors.

```
< optional error data >    none.
```

Section 2

Tests the interface between the Channel Data register and CYBER 170 channel receivers and transmitters using the CYBER 170 adapter Test mode facility and invert/force parity.

Specific hardware tested includes:

- Output Buffer Data mux, KZ pak
- CYBER 170 adapter output buffer, KZ pak
- CYBER 170 adapter output buffer parity checker, KZ pak
- CYBER 170 adapter output buffer parity generator, KZ pak
- CYBER 170 channel drivers, KZ pak
- CYBER 170 channel receivers, KZ pak
- CYBER 170 adapter input buffer, KZ pak
- CYBER 170 adapter input buffer parity checker, KZ pak

Subsection 0

Tests the PP to CYBER 170 channel interface in the CYBER 170 adapter.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Place a data pattern on the channel and read.

The Channel Error Flag, and the data input from the channel are checked for errors. This subsection consists of 14 conditions with each condition being a different data pattern.

```
< optional error data >    PATTERN = pppp
```

Subsection 1

Tests the parity bit for the Transmit register in the CYBER 170 DMA adapter.

Method of Execution;

1. Place the CYBER 170 adapter in Test mode.
2. Enable the Invert Transmit Parity Test mode facility (force error code 12) in the CYBER 170 adapter.
3. Place a data pattern on the channel and read.

The Error Status register, Channel Error Flag, and the data input from the channel are checked for errors. This subsection consists of two conditions with each condition being a different data pattern.

```
< optional error data >    PATTERN = pppp
```

Subsection 2

Tests the parity for the channel input data in the CYBER 170 DMA adapter.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Enable the Invert Channel Input Data Parity Test mode facility (force error code 13) in the CYBER 170 adapter.
3. Place a data pattern on the channel and read.

The Error Status register, Channel Error Flag, and the data input from the channel are checked for errors. This subsection consists of two conditions with each condition being a different data pattern.

```
< optional error data >    PATTERN = pppp
```

Subsection 3

Tests the CYBER 170 output buffer parity generator in the CYBER 170 adapter.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Place a data word on the channel and read.

The Channel Error Flag and the data input from the channel are checked for errors. This subsection consists of 256 conditions with each condition being a different data pattern.

```
< optional error data >    PATTERN = pppp
```

Subsection 4

Tests the CYBER 170 input buffer in the CYBER 170 adapter.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Output a block of 10 data words on the channel and read, followed by a two-word output and a two-word input.

The Channel Error Flag and the data input from the channel are checked for errors. This subsection consists of 12 conditions with each condition being a different block of 12 data words.

```
< optional error data >    none.
```

Subsection 5

Tests the CYBER 170 adapter input buffer overflow response from the CYBER 170 adapter.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Enable the Enable Overflow test.
3. Output a block of 10 data words on the channel.
4. Input a block of 9 data words from the channel.

The Error Status register, Channel Error Flag, and the data input from the channel are checked for errors.

```
< optional error data >     none.
```

Section 3

This section is the test/external clock test.

Subsection 0

Tests the enable test clock.

Method of Execution:

1. Read and save the Operational Status register.
2. Place the CYBER 170 adapter in Test mode.
3. Enable the test clock. If the external clock status bit is not set in the Adapter Operational Status word, the Clock Fault bit is checked for being set in the Error Status register.
4. Read and check the Operational Status register for errors.
5. Place a data pattern on the channel and read.

The Channel Error Flag and the data input from the channel are checked for errors. This subsection consists of 14 conditions with each condition being a different data pattern.

```
< optional error data >    PATTERN = pppp
```

Subsection 1

Tests the Disable External Clock.

Method of Execution:

1. Read and save the Operational Status register.
2. Place the CYBER 170 adapter in Test mode.
3. Disable the external clock. If the External clock bit was not set in the Operational Status register, the Clock Fault bit is not expected to be set in the Adapter Error Status register.
4. Read and check the Error Status register for errors.
5. Place a data pattern on the channel and read.

The Channel Error Flag and the data input from the channel is checked for errors. This subsection consists of 14 conditions with each condition being a different data pattern.

```
< optional error data >    PATTERN = pppp
```

Section 4

This section tests PP transfer.

Subsection 0

Tests the channel flag status bit from the CYBER 170 Adapter in the IOU Maintenance B register.

Method of Execution:

1. Set the channel flag on the CIO channel.
2. Read the IOU Maintenance B register.
3. Check the channel flag status bit for being set.
4. Clear the channel flag on the CIO channel.

The IOU Maintenance B register is read and the channel flag status bit checked for being clear.

```
< optional error data >    none.
```

Subsection 1

Tests the Full I, Full II, Full I or Full II, and Active Channel Flag status bits from the CYBER 170 adapter in the IOU Maintenance B register.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Output one PP word on the adapter channel.
3. Check the Active, Full I or Full II, and Full II Channel Flag status bits for being set in the IOU Maintenance B register for the CYBER 170 adapter.
4. Output a block of 10 PP words on the adapter channel.
5. Check the Active, Full I or Full II, and Full II and Full I Channel Flag status bits for being set in the IOU maintenance B register for the CYBER 170 adapter.

The data is input from the adapter channel.

```
< optional error data >    none.
```

Subsection 2

Tests PP transfers.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Generate a random data pattern for the test condition. One hundred and sixty data words are output/input from the channel.

The Channel Error Flag and the data input from the channel are checked for errors. This subsection consists of 10 conditions with each condition being a different random data pattern.

```
< optional error data >

TEST SEED ss
XFER BLOCK  xxxx xxxx xxxx xxxx
```

Section 5

This section tests the DMA Transfer registers, using the Clear T register, Read T register, and Write T register functions and force parity.

Specific hardware tested includes:

- T registers driver, JY pak
- T register, JY pak
- T register, JY pak
- T register 3-1 output mux, JY pak
- T register byte count parity checker, JY pak
- T register parity checker, JY pak
- Operational Status register, KX pak

Subsection 0

Tests the Write T register function.

Method of Execution:

1. Issue a Write T register function.
2. Output three words.
3. Attempt to output a fourth word.

The channel signal response and the Error Status register are checked for errors.

```
< optional error data >    none.
```

Subsection 1

Tests the Clear T register function.

Method of Execution:

1. Load the T register with a data pattern.
2. Issue a Clear T register function.

The contents of the T register is read and verified to be zero, and the Error Status register is checked for errors.

```
< optional error data >
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 2

Tests the interface between the T registers and the Channel Data register.

Method of Execution:

1. Issue the Clear T register function.
2. Issue the Write T register function to load data into the T register.
3. Read the T register using the Read T register function.

The Error Status register and the contents for the T register are checked for errors. This subsection consists of 45 conditions with each condition being a different data pattern.

```
< optional error data >
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 3

Tests the T register parity checkers.

Method of Execution:

1. Master clear the CYBER 170 adapter.
2. Enable the Force T register Parity (force error code 22).
3. Load a data pattern into the T register.

The Error Status register and Channel Error Flags are checked for errors. This subsection consists of seven conditions with each condition being a different data pattern.

```
< optional error data >
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 4

Test the T register parity generators.

Method of Execution:

1. Load the T register with a data pattern.
2. Read the contents of the T register.

The Channel Error Flag, and the contents of the T register are checked for errors. This subsection consists of 30 conditions with each condition being a different data pattern.

```
< optional error data >
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 5

Test the T register empty status.

Method of Execution:

1. Issue a Clear T register function.

2. Load the T register with a data pattern.

The Operational Status register is checked for errors, the T register is loaded with the same data pattern, and the Operational Status register is again checked for errors.

```
< optional error data >
XFER BLOCK cccc bbbb tttt tttt
```

Section 6

This section tests the PP Word Counter (PWC) in the CYBER 170 adapter.

Specific hardware tested includes:

- T register, JY pak
- T register, JY pak
- Operational Status register, KX pak
- PP Word Counter, KX pak

Subsection 0

Tests that the CYBER 170 adapter PWC is loaded correctly.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Issue the Start DMA Input function to load the PWC register with a data pattern.
3. Read the Channel Status register in the Maintenance register.

The PWC, Operational, and Error Status are checked for errors in the Channel Status register. This subsection consists of 18 conditions with each condition being a different data pattern.

```
< optional error data >    PATTERN = xxxx
```

Subsection 1

Tests that the CYBER 170 adapter PWC, decrements properly to zero using a PP word count of one, and enters DMA mode after decrementing to zero.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Load the T register with the central memory (CM) address and byte count of zero.
3. Initiate a DMA Input transfer with a PWC of one.
4. Output and input one PP word.
5. Read the Channel Status register in the Maintenance register.

The PWC, Operational, and Error Status are checked for errors in the Channel Status register.

```
< optional error data >    PATTERN = xxxx
```

Subsection 2

Tests that the CYBER 170 adapter PWC, decrements properly to zero using a PP word count of 255 PP words, and enters DMA mode after decrementing to zero.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Load the T register with the CM address and byte count of zero.
3. Initiate a DMA Input transfer with a PWC of 255.
4. Output and input 255 PP words.
5. Read the Channel Status register in the Maintenance register.

The PWC, operational and error status are checked for errors in the Channel Status register.

```
< optional error data >    none.
```

Section 7

This section tests the CM to CYBER 170 adapter Transfer register interface using the Test mode facility in DMA Output mode and invert parity. Specific hardware tested includes:

- Output buffer data mux, KZ pak
- Disassembly buffer, JY pak
- Disassembly output buffer, JY pak
- Disassembly buffer parity checker, JY pak
- T register, JY pak

Subsection 0

Tests CYBER 170 adapter Transfer-In-Progress status bit in the Operational Status register.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode with inhibit full out set.
2. Initialize the CM buffer with a unique data pattern.
3. Load the T register with the CM address and byte count of eight (one CM word).
4. Initiate a DMA Output transfer with a PWC of zero using 60-bit CM words.

The Operational Status register is checked for errors to insure that Transfer-In-Progress and output buffer Full are set. Then, Inhibit Full Out is disabled in the CYBER 170 adapter to allow the transfer to complete. The Operational Status register is checked for errors to insure that Input Data Available is set. Five PP words are input by the PP. The data input from the channel is checked for errors.

```
< optional error data >
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
DATA PTRN pppp pppp pppp pppp
```

Subsection 1

Tests the CM to CYBER 170 adapter transfer register interface.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Initialize the CM buffer with the data pattern.
3. Load the T register with the CM address and byte count of eight (one CM word).
4. Initiate a DMA output transfer with a PWC of zero using 60-bit CM words.
5. Input five PP words by the PP.

The NIO Fault Status 1 and Error Status registers are checked for errors. This subsection consists of 13 conditions with each condition being a different data pattern. The real memory address and byte count remain unchanged for each condition.

```
< optional error data >
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
DATA PTRN pppp pppp pppp pppp
```

Subsection 2

Tests the Disassembly Buffer ranks.

Method of Execution;

1. Place the CYBER 170 adapter in Test mode with Inhibit Full Out enabled.
2. Initialize the CM buffer with the data pattern.
3. Load the T register with the CM address and byte count for 160 (20 CM words).
4. Initiate a DMA output transfer with a PWC of zero using 60-bit CM words.

When the transfer stops, Inhibit Full Out is disabled allowing the transfer to complete. The Operational Status register is checked for errors to insure that Input Data Available and input buffer Full are set. One-hundred PP words are input by the PP. The NIO Fault Status 1 and Error Status registers are checked for errors. This subsection consists of 13 conditions with each condition being a different data pattern. The real memory address and byte count remain unchanged for each condition.

```
< optional error data >
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
DATA PTRN pppp pppp pppp pppp
```

Subsection 3

Tests the disassembly buffer parity generators.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Initialize the CM buffer with the data pattern.
3. Load the T register with the CM address and byte count of eight (one CM word).
4. Initiate a DMA output transfer with a PWC of zero using 60-bit CM words.
5. Input five PP words by the PP.

The NIO Fault Status 1 and Error Status registers are checked for errors. This subsection consists of eight conditions with each condition being a different data pattern. The real memory address and byte count remain unchanged for each condition.

```
< optional error data >
TEST SEED ss XFER BLOCK cccc bbbb tttt tttt
DATA PTRN pppp pppp pppp pppp
```

Subsection 4

Tests the disassembly buffer lower byte parity checker.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode with Invert Lower Adapter Output Parity Bit (force error code 24) enabled.
2. Initialize the CM buffer.
3. Load the T register with the CM address and byte count of eight (one CM word).
4. Initiate a DMA Output transfer with a PWC of zero using 60-bit CM words.
5. Input five PP words by the PP.

The NIO Fault Status 1, CIO Fault Status 2, and Error Status registers are checked for errors.

```
< optional error data >    none.
```

Subsection 5

Tests the disassembly buffer upper byte parity checker.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode with Invert Upper Adapter Output Parity Bit (force error code 23) enabled.
2. Initialize the CM buffer.
3. Load the T register with the CM address and byte count of eight (one CM word).
4. Initiate a DMA output transfer with a PWC of zero using 60-bit CM words.
5. Input five PP words by the PP.

The NIO Fault Status 1, CIO Fault Status 2, and Error Status registers are checked for errors.

```
< optional error data >    none.
```

Section 8

This section is the CYBER 170 adapter channel backup test. The Test mode facility in PP Output mode is used.

Specific hardware tested includes:

- CYBER 170 adapter output buffer, KZ pak

Subsection 0

Tests the CYBER 170 adapter output buffer.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode with Inhibit Full Out enabled.
2. Initialize the memory buffer with the data pattern.
3. Load the T register with the CM address and byte count of 120 (15 CM words).
4. Initiate a DMA output transfer with a PWC of zero using 60-bit CM words.
5. Input 75 PP words by the PP.
6. Disable the Inhibit Full Out.
7. Input 75 PP words by the PP.

The Channel Error Flag is checked for errors. This subsection consists of 18 conditions with each condition being a different data pattern.

```
< optional error data >
PATTERN = nnnn
```

Section 9

This section tests the CYBER 170 adapter Transfer register CM interface using the Test mode facility in DMA Input mode and invert parity. Specific hardware tested includes:

- Input Data Fan-In, KZ pak
- Assembly buffer, JY pak
- DMA input data buffer, KZ pak
- Assembly output buffer, JY pak

Subsection 0

Tests the ADU Transfer-In-Progress. This subsection is bypassed if PARAM15 is set.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Generate a unique data pattern in the PP.
3. Block CM requests at the CMI.
4. Load the T register with the CM address and byte count of eight (one CM word).
5. Initiate a DMA input transfer with a PWC of zero using 60-bit CM words.
6. Output five PP words by the PP.

The Operational Status register is checked for errors to ensure that Transfer-In-Progress is set. Then, CM requests are released at the CMI to allow the transfer to complete. Data in CM is checked for errors.

```
< optional error data >
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 1

Tests the CYBER 170 adapter Transfer register CMI.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Generate a data pattern in the PP.
3. Load the T register with the CM address and byte count of eight (one CM word).
4. Initiate a DMA input transfer with a PWC of zero using 60-bit CM words.
5. Output five PP words by the PP.

Data in CM, NIO Fault Status 1, and Error Status registers are checked for errors. This subsection consists of 13 conditions with each condition being a different data pattern. The real memory address and byte count remain unchanged for each condition.

```
< optional error data >
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 2

Tests the assembly buffer ranks. This subsection is bypassed if PARAM15 is set.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Block CM requests at CMI.
3. Generate a data pattern in the PP.
4. Load the T register with the CM address and byte count for a 16 (two CM words).
5. Initiate a DMA input transfer with a PWC of zero using 60-bit CM words.
6. Output 79 PP words by the PP.

Operational Status is checked for errors. Central memory requests are then allowed to CM allowing the transfer to complete. The NIO Fault Status 1, Error Status registers, and data in CM are checked for errors. This subsection consists of 13 conditions with each condition being a different data pattern.

```
< optional error data >
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 3

Tests the assembly buffer parity generators.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Generate a data pattern in the PP.
3. Load the T register with the CM address and byte count for 64 (eight CM words).
4. Initiate a DMA input transfer with a PWC of zero using 60-bit CM words.
5. Output 325 PP words by the PP.

The NIO Fault Status 1, Error Status registers, and data in CM are checked for errors. This subsection consists of four conditions with each condition being a different data pattern.

```
< optional error data > TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 4

Tests the assembly buffer lower byte parity checker.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode with Force Adapter Input Parity Bit 0 Low (force error code 21) enabled.
2. Generate a data pattern in the PP.
3. Load the T register with the CM address and byte count of eight (one CM word).
4. Initiate a DMA input transfer with a PWC of zero using 60-bit CM words.
5. Output five PP words by the PP.

The NIO Fault Status 1, Error Status registers, and data in CM are checked for errors. This subsection consists of three conditions with each condition being a different data pattern.

```
< optional error data >
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 5

Tests the assembly buffer upper byte parity checker. Same as subsection 4 only Force Adapter Input Parity Bit 1 Low (force error code 20) is used.

```
< optional error data >
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Section 10

This section tests the DMA transfer register address incrementer using the Test mode facility in DMA Output mode.

Specific hardware tested includes:

- T register real memory address incrementer, JY pak
- T register parity prediction network, JY pak

Subsection 0

Tests the DMA transfer register address incrementer.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Generate a data pattern in the PP.
3. Load the T register with the CM address pattern and byte count of eight (one CM word).
4. Initialize the CM address that will be read if the address is within the defined CM buffer boundaries.
5. Initiate a DMA output transfer with a PWC of zero using 60-bit CM words.
6. Input five PP words by the PP.

The NIO Fault Status 1, Error Status registers, and T register are read and checked. This subsection consists of 56 conditions with each condition being a different CM address pattern.

```
< optional error data >
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 1

Tests force address parity prediction error.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode with Invert Address Parity Prediction (force error code 25) enabled.
2. Generate a data pattern in the PP.
3. Load the T register with the CM address and byte count of 16 (two CM words).
4. Initiate a DMA input transfer with a PWC of zero using 60-bit CM words.
5. Output 10 PP words by the PP.

The NIO Fault Status 1, Error Status registers, and data in CM are checked for errors.

```
< optional error data >
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Section 11

This section tests the DMA transfer register byte counter decrementer using the Test mode facility in DMA Output mode.

Specific hardware tested includes:

- T register byte count decrementer, JY pak

Subsection 0

Tests the DMA transfer register byte counter decrementer.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Initialize the CM buffer with a unique data pattern.
3. Load the T register with the CM address and maximum byte count available.
4. Initiate a DMA output transfer with a PWC of zero using 60-bit CM words.
5. Input the data pattern by the PP.

The NIO Fault Status 1 and Error Status registers are checked for errors.

```
< optional error data >
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Section 12

This section is the DMA transfer test.

Subsection 0

Tests incomplete word DMA output transfers in 60-bit mode.

Method of Execution:

1. Generate a data pattern in the PP.
2. Initialize the CM addresses that will be read with the data pattern.
3. Load the T register with the CM address pattern and byte count of 10 (one CM word plus).
4. Initiate a DMA output transfer with a PWC of zero using 60-bit CM words.
5. Input six PP words by the PP.

The NIO Fault Status 1, Error Status registers, and data read from CM are checked for errors.

```
< optional error data >
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 1

Tests incomplete word DMA output transfers in 64-bit mode.

Method of Execution:

1. Initialize the CM addresses that will be read with the data pattern.
2. Load the T register with the CM address pattern and byte count of 10 (one CM word plus).
3. Initiate a DMA output transfer with a PWC of zero using 64-bit CM words.
4. Input six PP words by the PP.

The NIO Fault Status 1, Error Status registers, and data read from CM are checked for errors.

```
< optional error data >
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 2

Tests incomplete word DMA input transfers in 60-bit mode.

Method of Execution:

1. Generate a data pattern in the PP.
2. Load the T register with the CM address pattern and byte count of 10 (one CM word plus).
3. Initiate a DMA input transfer with a PWC of zero using 60-bit CM words.
4. Output six PP words by the PP.

The NIO Fault Status 1, Error Status registers, and data in CM are checked for errors.

```
< optional error data >
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 3

Tests incomplete word DMA input transfers in 60-bit mode.

Method of Execution:

1. Generate a data pattern in the PP.
2. Load the T register with the CM address pattern and byte count of 10 (one CM word plus).
3. Initiate a DMA input transfer with a PWC of zero using 64-bit CM words.
4. Output six PP words by the PP.

The NIO Fault Status 1, Error Status registers, and data in CM are checked for errors.

```
< optional error data >
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 4

Tests DMA transfers in 60-bit mode.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Generate a random data pattern in the PP.
3. Load the T register with the CM address pattern and random byte count for the number of CM words within the defined CM boundaries.
4. Initiate a DMA input transfer with a PWC of zero using 60-bit CM words.
5. Output the random data pattern by the PP for the DMA input transfer.
6. Reload the T registers with the same values used for the DMA input transfer.
7. Initiate a DMA output transfer with a PWC of zero using 60-bit CM words.
8. Input the data by the PP for the DMA output transfer.

The NIO Fault Status 1, Error Status registers, and DMA output date are checked for errors. This subsection consists of 10 conditions with each condition being a different random data pattern and random byte count.

```
< optional error data >
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Section 13

This section is the DMA transfer error test. Errors are forced in this section by using the DMA adapter's hardware functions.

Subsection 0

Tests response on invert 12/16 shifter parity.

Method of Execution:

1. Invert the parity for the CB array at the generator is forced (force error code 10).
2. Place the CYBER 170 adapter in Test mode.
3. Initialize CM with the data pattern.
4. Load the T register with the CM address pattern and byte count of eight (one CM word).
5. Initiate a DMA output transfer with a PWC of zero using 60-bit CM words.
6. Input five PP words by the PP.

The Error Status register, CIO Fault Status 2 register and data input from CM are checked for errors.

```
< optional error data > TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 1

Tests response on invert conversion parity error.

Method of Execution:

1. Invert the parity bit on the converted data at the checker is forced (force error code 11).
2. Place the CYBER 170 adapter in Test mode.
3. Initialize the CM buffer with the data pattern.
4. Load the T register with the CM address pattern and byte count of eight (one CM word).
5. Initiate a DMA output transfer with a PWC of zero using 60-bit CM words.
6. Input five PP words by the PP.

The Error Status register, CIO Fault Status 2 register, and data input from CM are checked for errors.

```
< optional error data >  none.
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 2

Tests response on invert CM response code parity error.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Invert the parity of the CM response code is forced (force error code 05).
3. Generate a data pattern in the PP.
4. Load the T register with the CM address pattern and byte count of eight (one CM word).
5. Initiate a DMA input transfer with a PWC of zero using 60-bit CM words.
6. Output five PP words by the PP.

The Error Status register, NIO Fault Status 1 register, CIO Fault Status 2 register, and data in CM are checked for errors.

```
< optional error data >
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 3

Tests response on invalid response code.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Force an illegal translation of the CM response code is enabled (force error code 04).
3. Generate a data pattern in the PP.
4. Load the T register with the CM address pattern and byte count of eight (one CM word).
5. Initiate a DMA input transfer with a PWC of zero using 60-bit CM words.
6. Output five PP words by the PP.

The Error Status register, NIO Fault Status 1 register, CIO Fault Status 2 register, and data in CM are checked for errors.

```
< optional error data >
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 4

Tests response on force byte count equal to zero (force error code 26).

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Force the CM request counter to report an error after 16 CM requests have been issued.
3. Initialize the CM buffer.
4. Load the T register with the CM address pattern and byte count of 160 (20 CM words).
5. Initiate a DMA output transfer with a PWC of zero using 60-bit CM words.
6. Input five PP words by the PP.

The Error Status register, NIO Fault Status 1 register, CIO Fault Status 2 register, and data input from CM are checked for errors.

```
< optional error data >
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 5

Tests the parity bit for the transmit register in the CYBER 170 DMA adapter.

1. Enable the Invert Transmit Parity Test Mode Facility (force error code 12) in the CYBER 170 adapter.
2. Place a data pattern on the channel and read.

The Error Status register, Channel Error Flag and the data input from the channel are checked for errors. This subsection consists of two conditions with each condition being a different data pattern.

```
< optional error data >    none.
```

Subsection 6

Test false start transfer, byte count zero.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Initialize the CM buffer with the PP unique data pattern.
3. Load the T register with the CM address pattern and byte count of zero.
4. Initiate a DMA input transfer with a PWC of zero using 60-bit CM words.

The Error Status register, NIO Fault Status 1 register, CIO Fault Status 2 register, and data in CM are checked for errors.

```
< optional error data >
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Section 14

This section is the DMA transfer error test. Errors are forced in this section by utilizing the Test mode facilities in the IOU Maintenance register. This section is bypassed if PARAM15 is set.

NOTE

Subsections 0, 3, 4, 5, and 6 are not executed when in concurrent maintenance mode.

Subsection 0

Test response on uncorrected write error. Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Enable Force Data-Out Parity function at the CMI.
3. Generate a data pattern in the PP.
4. Load the T register with the CM address pattern and byte count of four.
5. Initiate a DMA input transfer with a PWC of zero using 60-bit CM words.
6. Output five PP words by the PP.

The Error Status register, NIO Fault Status 1 register, and CIO Fault Status 2 register are checked for errors.

```
< optional error data >
MR TEST MODE =mm
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 1

Test response on uncorrected read error. Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Enable Force Data-In Parity function at the CMI.
3. Load T register with CM address pattern and byte count of eight (one CM word).
4. Initiate a DMA output transfer with a PWC of zero using 60-bit CM words.
5. Input five PP words by the PP.

The Error Status register, CIO Channel Status register, NIO Fault Status 1 register, and CIO Fault Status 2 register are checked for errors.

```
< optional error data >
MR TEST MODE =mm
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 2

Test response on CMI response code parity error.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Enable Force Response Code Parity Error function at the CMI.
3. Generate a data pattern in the PP.
4. Load the T register with the CM address pattern and byte count of eight (one CM word).
5. Initiate a DMA input transfer with a PWC of zero using 60-bit CM words.
6. Output five PP words by the PP.

The Error Status register, NIO Fault Status 1 register, and CIO Fault Status 2 register are checked for errors.

```
< optional error data >
MR TEST MODE =mm
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 3

Test response on CMI reject.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Enable Invert Function Out Parity function at the CMI.
3. Initialize the CM buffer with the data pattern.
4. Load the T register with the CM address pattern and byte count of eight (one CM word).
5. Initiate a DMA output transfer with a PWC of zero using 60-bit CM words.
6. Input five PP words by the PP.

The Error Status register, CIO Channel Status register, NIO Fault Status 1 register, and CIO Fault Status 2 register are checked for errors.

```
< optional error data >
MR TEST MODE =mm
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 4

Test response on address out parity error.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Enable Address Out Parity function at the CMI.
3. Initialize the CM buffer with the data pattern.
4. Load the T register with the CM address pattern and byte count of eight (one CM word).
5. Initiate a DMA output transfer with a PWC of zero using 60-bit CM words.
6. Input five PP words by the PP.

The Error Status register, CIO Channel Status register, NIO Fault Status 1 register, and CIO Fault Status 2 register are checked for errors. This subsection is repeated for an address bit in each parity byte.

```
< optional error data >
MR TEST MODE =mm
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 5

Test response on tag-in/out errors.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Enable Force Tag-In/Out Parity Error function at the CMI.
3. Generate a data pattern in the PP or
4. Initialize the CM buffer with the data pattern, as determined by the type of DMA transfer.
5. Load the T register with the CM address pattern and byte count of eight (one CM word).
6. Initiate a DMA transfer with a PWC of zero using 60-bit CM words.
7. Output five PP words by the PP if the DMA transfer type is a DMA input transfer.

The Error Status register, NIO Fault Status 1 register, and CIO Fault Status 2 register are checked for errors. This subsection is repeated for transfers in both directions.

```
< optional error data >
MR TEST MODE =mm
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 6

Test response on mark parity bit errors.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Enable Force Mark Parity Bit to Zero function at CMI.
3. Generate a data pattern in the PP or
4. Initialize the CM buffer with the data pattern, as determined by the type of DMA transfer.
5. Load the T register with the CM address pattern and byte count of eight (one CM word).
6. Initiate a DMA transfer with a PWC of zero using 60-bit CM words.
7. Output or input five PP words by the PP as determined by the type of DMA transfer.

The Error Status register, NIO Fault Status 1 register, and CIO Fault Status 2 register are checked for errors. This subsection is repeated for transfers in both directions.

```
< optional error data >
MR TEST MODE =mm
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Section 15

This section is the PWC/DMA transfer test. Input and output DMA data transfers are performed using the PWC set to a nonzero value.

Subsection 0

Tests DMA output transfers in 60-bit mode.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Initialize the CM buffer with the random data pattern.
3. Load the T register with the CM address pattern and byte count for 16 (two CM words).
4. Initiate a DMA output transfer with a PWC of five PP words using 60-bit CM words.
5. Output and input five PP words by the PP.
6. Input the random data pattern by the PP for the DMA output transfer.

The NIO Fault Status 1, Error Status registers, PWC data, and data input from CM are checked for errors.

```
< optional error data >
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 1

Tests DMA input transfers in 60-bit mode.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Generate a random data pattern in the PP.
3. Load the T register with the CM address pattern and byte count for 16 (two CM words).
4. Initiate a DMA input transfer with a PWC of five PP words using 60-bit CM words.
5. Output and input five PP words by the PP.
6. Output the random data pattern by the PP for the DMA input transfer.

The NIO Fault Status 1, Error Status registers, PWC data, and data in the CM buffer are checked for errors.

```
< optional error data >
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 2

Tests DMA Output transfers in 64-bit mode.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Initialize the CM buffer with the random data pattern.
3. Load the T register with the CM address pattern and byte count for 16 (two CM words).
4. Initiate a DMA output transfer with a PWC of five PP words using 64-bit CM words.
5. Output and input five PP words by the PP.
6. Input the random data pattern by the PP for the DMA output transfer.

The NIO Fault Status 1, Error Status registers, PWC data, and data input from CM are checked for errors.

```
< optional error data >
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Subsection 3

Tests DMA input transfers in 64-bit mode.

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Generate a random data pattern in the PP.
3. Load the T register with the CM address pattern and byte count for 16 (two CM words).
4. Initiate a DMA input transfer with a PWC of five PP words using 64-bit CM words.
5. Output and input five PP words by the PP.
6. Output the random data pattern by the PP for the DMA input transfer.

The NIO Fault Status 1, Error Status registers, PWC data, and data in the CM buffer are checked for errors.

```
< optional error data >
TEST SEED ss
XFER BLOCK cccc bbbb tttt tttt
```

Section 16

This section is the stacked transfer test. A minimum of six transfers is sequentially sent to the CYBER 170 adapter.

Subsection 0

Method of Execution:

1. Place the CYBER 170 adapter in Test mode.
2. Generate a random data pattern in the PP. The byte counts are generated randomly. The starting real memory addresses are computed from the byte counts.
3. Initiate DMA input transfers with a PWC of zero using 60-bit CM words until the byte count/address pairs have expired.
4. Initiate DMA output transfers with a PWC of zero using 60-bit CM words with the same byte count/address pairs.

The NIO Fault Status 1, Error Status registers, and data input by the PP are then checked for errors. This sequence is repeated for ten conditions.

```
< optional error data >
TEST SEED ss
XFER NO. nn
NXT cccc bbbb tttt tttt
CUR cccc bbbb tttt tttt
```

IPI4/IPI42/IPI44/IPI46 DMA-Enhanced IPI Channel Adapter 7

IPI4/IPI42/IPI44/IPI46 DMA-Enhanced IPI Channel Adapter 7

Section Descriptions

Section 00 – IPI Adapter PP Access Test

This section tests the IPI adapter access from the peripheral processor (PP). Channel signal response is verified by using the Master Clear Adapter function, Clear Error function, Illegal functions, Read Error Status function with channel input/output (I/O), and the Invert PP to Channel Parity facility.

Subsection 0

Tests the response on Master Clear Adapter function.

Method of Execution:

- Issue a Master Clear Adapter (0000_{16}) function to the IPI adapter.

All MAC IPI adapter Status register bytes except the Operational Status register are checked to be zero.

```
( optional error data ) none.
```

Subsection 1

Tests the response on Clear Error function.

Method of Execution:

- Issue a Clear Error (0100_{16}) function to the IPI adapter.

The MAC IPI adapter Status register's DMA Error register status and IPI Error register status bytes are checked to be zero.

```
( optional error data ) none.
```

Subsection 2

Tests the response on Read Error Status register function.

1. Issue a Read Error Status register (0600_{16}) function to the IPI adapter.
2. Read and check the DMA Error register status to be zero. The CIO Fault Status 2 register is also checked.

```
( optional error data ) none.
```

Subsection 3

Tests the response on Illegal function.

Method of Execution:

1. Issue several illegal functions to the IPI adapter.
2. Read and check the DMA Error register status for Illegal function or sequence status.
3. Check the MAC IPI adapter Status registers DMA Error register status Byte 0 for Illegal function or sequence status. The CIO Fault Status 2 register is also checked.

```
( optional error data ) none.
```

Subsection 4

Tests the Channel Data register parity checker.

Method of Execution:

1. Enable the maintenance channel's Invert PP to Channel Parity at Generator (I4 Test mode Maintenance register test code 02B).
2. Place a data pattern on the channel and read.
3. Check the DMA error status and CIO Fault Status 2 registers for errors.
4. Check the MAC IPI adapter Status register's Flag Status register Byte 5 for error flag status.
5. Check the MAC IPI adapter Status register's DMA Error register status Byte 1 for BAS parity error and LX error status.

This subsection consists of four conditions, each being a different data pattern. The subsection is not to be executed in multiprocessing mode (see PARAM15).

```
( optional error data )  PATTERN = pppp
```

Subsection 5

Tests the response on function parity error.

Method of Execution:

1. Enable the maintenance channel's Invert PP to Channel Parity at Generator (I4 Test mode Maintenance register test code 02B).
2. Issue a function to the IPI adapter.
3. Check the DMA error status and CIO Fault Status 2 registers.
4. Check the MAC IPI adapter Status register's Flag Status register Byte 5 for error flag status.
5. Check the MAC IPI adapter Status register's DMA Error register status Byte 1 for BAS parity error and LX error status.

This subsection is not executed in multiprocessing mode (see PARAM15).

```
( optional error data ) none.
```

Section 01 – Control Register Test

This section tests the IPI adapter Control register utilizing the Write Control register and Read Control register functions and the Invert Control register parity facility.

Subsection 0

Tests the interface between the Channel Data register and Control register.

Method of Execution:

1. Load data into the Control register using the Write Control register (0300_{16}) function.
2. Issue a Read Control register (0200_{16}) function and read data.
3. Check the MAC IPI adapter Status register's DMA Control register Byte 6 for the appropriate bits being set.
4. Check the DMA Error register status for errors.

This subsection consists of 17 conditions, each being a different pattern.

```
( optional error data )  CONTROL REGISTER = cccc
```

Subsection 1

Tests the Control register parity generator.

Method of Execution:

1. Load data into the Control register.
2. Read the Control register and compare data.
3. Check the DMA Error register status for errors.

This subsection consists of 256 conditions each being a different pattern.

```
( optional error data )  CONTROL REGISTER = cccc
```

Subsection 2

Tests that the Master Clear Adapter function clears the Control register.

Method of Execution:

1. Load nonzero data into the Control register.
2. Issue a Master Clear Adapter function.
3. Read the register and check data to be zero.

```
( optional error data ) none.
```

Subsection 3

Tests the Control register force error codes that statically produce errors (that is, those error codes that do not require data transfers or other actions).

Method of Execution:

1. Load the Control register with one of the force error codes listed below.
2. Read the DMA Error register status and checked for the appropriate response.
3. Check Fault Status 2.
4. Clear the Control register and read the DMA Error register status twice to insure that the second read returns zeros.
5. Check the MAC IPI adapter Status register for the appropriate status.

```
( optional error data )  CONTROL REGISTER = cccc

FORCE CONTROL REGISTER PARITY ERROR  (ERROR
FORCE  CODE  17(16)) (MAC byte 1, DMA REGISTER
PE and LZ ERROR)
FORCE PP BAS INPUT DATA PARITY ERROR (ERROR
FORCE  CODE  09(16)) (MAC byte 1, DMA REGISTER
PE and LZ ERROR)
FORCE DMA/IPI INPUT DATA PARITY ERROR UPPER
(ERROR  FORCE  CODE  0A(16))  (MAC byte 1, DMA
REGISTER PE and LZ ERROR)
FORCE DMA/IPI INPUT DATA PARITY ERROR LOWER
(ERROR  FORCE  CODE  0B(16))  (MAC byte 1, DMA
REGISTER PE and LZ ERROR)
FORCE DEADMAN TIMER  COUNTER  PARITY  ERROR
LOWER  (ERROR  FORCE CODE 0B(16)) (MAC byte 1,
DMA REGISTER PE and LZ ERROR)
```

Subsection 4

Tests DMA Error Status register parity.

Method of Execution:

1. Set Force Error Status Register Parity Upper (force error code $0E_{16}$) or Force Error Status Register Parity Lower (force error code $0F_{16}$) in the Control register force error code (one per condition).
2. Read and check the DMA Error register status and Fault Status 1 register.

```
( optional error data )  CONTROL REGISTER = cccc
```

Subsection 5

Test Force Function Register Parity Error.

Method of Execution:

1. Set Force Function Register Parity Error (force error code 03_{16}) in the Control register force error code.
2. Check the MAC IPI adapter Status register for the DMA Error register status bits DMA register parity error and LZ error.

```
( optional error data )  CONTROL REGISTER = cccc
```

Subsection 6

Test Force BAS Register Parity Error.

Method of Execution:

1. Set Force BAS Register Parity Error Upper (force error code 06_{16}) or Force BAS Register Parity Error Lower (force error code 07_{16}) in the Control register force error code (one per condition).
2. Output and input a word on IPI CIO channel.
3. Check the MAC IPI adapter Status register for the DMA Error register status bits DMA register parity error, BAS parity error, LZ error, and LX error.

```
( optional error data )  CONTROL REGISTER = cccc
```

Section 02 – Operational Register Status Test

This section tests the DMA enhanced IPI adapter Operational Status register.

Subsection 0

Tests the response on the Read Operational Status register function.

Method of Execution:

1. Issue a Read Operational Status register (0700_{16}) function to the IPI adapter.
2. Read and check the Operational Status register for the appropriate bits set.
3. Check the CIO Fault Status 2 register.

   (optional error data) none.

Subsection 1

Tests Invert Operational Status Parity Upper.

Method of Execution:

1. Set the Invert Operational Status Parity Upper (force error code 00_{16}) in the Control register force error code.
2. Read the Operational Status register.
3. Check the DMA Error register status.

   (optional error data) none.

Subsection 2

Tests Invert Operational Status Parity Lower.

Method of Execution:

1. Set the Invert Operational Status Parity Lower (force error code 01_{16}) in the Control register force error code.
2. Read the Operational Status register.
3. Check the DMA Error register status.

   (optional error data) none.

Subsection 3

Tests the parity error status bits in the upper byte of the Operational Status register.

Method of Execution:

1. Load the Control register with one of the force error codes listed after item 6.
2. Read and check the Operational Status registers.
3. Read and check the DMA Error register status for the appropriate response.
4. Check the Fault Status 2 register.
5. Clear the Control register and read the DMA Error register status twice to insure that the second read returns zeros.
6. Check the MAC IPI adapter Status register for the appropriate status as listed below.

```
( optional error data ) CONTROL REGISTER = cccc

FORCE CONTROL REGISTER PARITY ERROR  (ERROR
FORCE   CODE   17(16))   (Operational   Status
Register bit Control register Parity Error)
FORCE PP BAS INPUT DATA PARITY ERROR (ERROR
FORCE CODE 09(16)) (Operational
Status Register  bit  Data  Register  Input
Parity Error)
FORCE DMA/IPI INPUT DATA PARITY ERROR UPPER
(FORCE ERROR CODE 0A(16)) (Operational  Status
Register  bit  Data  Register Output Parity
Error)
FORCE DMA/IPI INPUT DATA PARITY ERROR LOWER
(ERROR  FORCE CODE 0B(16)) (Operational Status
Register bit Data  Register  Output  Parity
Error)
FORCE  DEADMAN  TIMER  COUNTER PARITY ERROR
LOWER (FORCE ERROR CODE  0B(16))  (Operational
Status  Register  bit Deadman Timer Counter
Parity Error)
```

Section 03 – T-Register Test

This section tests the DMA Enhanced IPI adapters T register operations using the Clear T register, Read T register, and Write T register functions and force parity.

Subsection 0

Tests the Write T register function.

Method of Execution:

1. Issue a Write T register ($0B00_{16}$) function and output three words.
2. Attempt to output a fourth word.
3. Check the channel signal response; that is, Full should remain set.
4. Check the DMA Error register status.

   ```
   ( optional error data ) none.
   ```

Subsection 1

Tests the T' register empty bit and T register Data Transfer functions.

Method of Execution:

1. Write the T register.
2. Check the state of the T' register Empty bit for being set in the Operational Status register and the MAC IPI adapter Status register's Status register Byte 4.
3. Write the T register again.
4. Check the state of the T' register Empty bit is for being clear in the Operational Status register and the MAC IPI adapter Status register's Operational Status register Byte 4.
5. Read the T register compare data with the data written. This is done for several data patterns.

   ```
   ( optional error data )  T-REGISTER bbbb tttt tttt
   ```

Subsection 2

Tests the Clear T register function.

Method of Execution:

1. Write both T registers.
2. Check the T' register Empty bit for the clear state.
3. Clear the T registers using the Clear T register function.
4. Read the T register and check data for being zeros.

```
( optional error data )  T-REGISTER bbbb tttt tttt
```

Subsection 3

Tests the interface between the T registers and the Channel Data register.

Method of Execution:

1. Load data into the T register using the Write T register ($0B00_{16}$) function.
2. Read the T register using the Read T register ($0A00_{16}$) function.
3. Check the data.
4. Check the DMA Error register status for errors.

This subsection consists of 45 conditions each being a different pattern.

```
( optional error data )  T-REGISTER bbbb tttt tttt
```

Subsection 4

Tests the T register parity checkers.

Method of Execution:

1. Master clear the IPI adapter.
2. Set the Force T register Data Parity Bit Low (force error code 12_{16}) in the Control register force error code.
3. Load a data pattern into the T register twice.
4. Check the DMA Error register status for JY error and DMA register parity error status.

This subsection consists of seven conditions each being a different data pattern.

```
( optional error data )  T-REGISTER bbbb tttt tttt
```

Subsection 5

Test the T register parity generators.

Method of Execution:

1. Load the T register with a data pattern.
2. Read the contents of the T register and the check data.
3. Check the DMA Error register status for errors.

This subsection consists of 30 conditions each being a different pattern.

```
( optional error data )  T-REGISTER bbbb tttt tttt
```

Subsection 6

Tests the T' register Empty status.

Method of Execution:

1. Set the T' register Empty bit in the Control register.
2. Issue a Clear T register ($0E00_{16}$) function.
3. Clear the CIO channel flag.
4. Load the T register with a data pattern.
5. Check the CIO channel flag and Operational Status register.
6. Check the MAC IPI adapter Status register's Flag Status register Byte 5 for Channel Flag status.
7. Clear the CIO channel flag.
8. Load the T register with the same data pattern.
9. Check the CIO channel flag and Operational Status register.

```
( optional error data )  T-REGISTER bbbb tttt tttt
```

Subsection 7

Tests response on T register parity error.

Method of Execution:

1. Set the Force T register R/W Counter Parity Error (force error code 02_{16}) in the Control register force error code.
2. Check the MAC IPI adapter Status register's Operational Status register Byte 4 and Operational Status register for T register parity error, PP/IPI mode and T' register Empty.
3. Check the MAC IPI adapter Status register's Error register status Byte 1 and DMA Error register status for DMA register parity error and LZ error.

```
( optional error data ) none.
```

Section 04 - CM to CYBER Channel Interface Test

This section tests the central memory (CM) to IPI adapter transfer register interface from the PP in DMA output mode and invert parity.

Subsection 0

Tests the IPI channel Transfer-in-Progress status bit in the Operational Status register.

Method of Execution:

1. Initialize the CM buffer with a unique data pattern.
2. Load the T register with the CM address and byte count of eight (one CM word).
3. Initiate a CM to PP data transfer.
4. Input one PP word is by the PP.
5. Check the DMA Transfer-In-Progress, output mode, PP/IPI mode, T' register empty, and Transfer-in-Progress status bits to insure that they are set in the MAC IPI adapter Status register's Operational Status register Byte 4.
6. Input three PP words by the PP.
7. Check the DMA Error register status.

```
( optional error data )  TEST SEED ssss
                         XFER BLOCK cccc bbbb tttt tttt
```

Subsection 1

Tests the CM to IPI adapter Transfer register interface.

Method of Execution:

1. Initialize the CM buffer with the data pattern.
2. Load the T register with the CM address and byte count of eight (one CM word).
3. Initiate a CM to PP data transfer.
4. Input four PP words by the PP.
5. Check the DMA Error register status and the NIO Fault Status 1 register for errors.
6. Check the data read from CM.

This subsection consists of 13 conditions with each condition being a different pattern. The real memory address and byte count remain unchanged for each condition.

```
( optional error data )  TEST SEED ssss
                         XFER BLOCK cccc bbbb tttt tttt
```

Subsection 2

Tests the disassembly buffer ranks.

Method of Execution:

1. Initialize the CM buffer with the data pattern.
2. Load the T register with the CM address and byte count of 160 (20 CM words).
3. Initiate a CM to PP data transfer.
4. Input 80 PP words by the PP.
5. Check the DMA Error register status and the NIO Fault Status 1 register for errors.
6. Check the data read from CM.

This subsection consists of 13 conditions with each condition being a different data pattern. The real memory address and byte count remain unchanged for each condition.

```
( optional error data )  TEST SEED ssss
                             XFER BLOCK cccc bbbb tttt tttt
```

Subsection 3

Tests the disassembly buffer parity generators.

Method of Execution:

1. Initialize the CM buffer with the data pattern.
2. Load the T register with the CM address and byte count of eight (one CM word).
3. Initiate a CM to PP data transfer.
4. Input four PP words by the PP.
5. Check the DMA Error register status and the NIO Fault Status 1 register errors.
6. Check the data read from CM.

This subsection consists of eight conditions with each condition being a different data pattern. The real memory address and byte count remain unchanged for each condition.

```
( optional error data )  TEST SEED ssss
                         XFER BLOCK cccc bbbb tttt tttt
                         DATA PTRN pppp pppp pppp pppp
```

Subsection 4

Tests the disassembly buffer lower byte parity checker.

Method of Execution:

1. Enable Invert Lower Channel Output Parity (force error code 14_{16}) in the IPI adapter Control register.
2. Initialize the CM buffer.
3. Load the T register with the CM address and byte count of eight (one CM word).
4. Initiate a CM to PP data transfer.
5. Input four PP words by the PP.
6. Check the DMA Error register status, NIO Fault Status 1 register and CIO Fault Status 2 register for errors.
7. Check the data read from CM.

(optional error data) none.

Subsection 5

Tests the disassembly buffer upper byte parity checker.

Method of Execution:

1. Enable the Invert Upper Channel Output Parity (force error code 13_{16}) in the IPI adapter Control register.
2. Initialize the CM buffer.
3. Load the T register with the CM address and byte count of eight (one CM word).
4. Initiate a CM to PP data transfer.
5. Input four PP words by the PP.
6. Check the DMA Error register status, NIO Fault Status 1 register and CIO Fault Status 2 register for errors.
7. Check the data read from CM.

(optional error data) none.

Section 05 – CYBER Channel To CM Interface Test

This section tests the IPI adapter to CM transfer register interface from the PP in DMA input mode and invert parity.

Subsection 0

Tests ADU Transfer-in-Progress.

Method of Execution:

1. Bypass this subsection if PARAM15 is set.
2. Generate a unique data pattern in the PP.
3. Block CM requests at the CMI.
4. Load the T register with the CM address and byte count of eight (one CM word).
5. Initiate a PP to CM data transfer.
6. Output four PP words by the PP.
7. Check the DMA Transfer-In-Progress, output mode, PP/IPI mode, T' register Empty, and Transfer-In-Progress status bits to ensure that they are set in the Operational Status register.
8. Release CM requests at the CMI to allow the transfer to complete.
9. Check data in CM for errors.

```
( optional error data )  TEST SEED ssss
                         XFER BLOCK cccc bbbb tttt tttt
```

Subsection 1

Tests TIP after terminated PP to CM DMA transfer.

Method of Execution:

1. Bypass this subsection if PARAM15 is set.
2. Generate a unique data pattern in the PP.
3. Block CM requests at the CMI.
4. Load the T register with the CM address and byte count of 16 (two CM words).
5. Initiate a PP to CM data transfer.
6. Output 64 PP words by the PP.
7. Release CM requests at the CMI to allow the transfer to complete.
8. Check the DMA Transfer-In-Progress, PP/IPI mode, T' register Empty, and Transfer-In-Progress status bits to ensure that they are set in the Operational Status Register.
9. Check the NIO Fault Status 1 and Error Status registers are checked for errors.

```
( optional error data )  TEST SEED ssss
                         XFER BLOCK cccc bbbb tttt tttt
```

Subsection 2

Tests the assembly buffer.

Method of Execution:

1. Generate a data pattern in the PP.
2. Load the T register with the CM address and byte count of eight (one CM word).
3. Initiate a PP to CM data transfer.
4. Output four PP words by the PP.
5. Check the DMA Error register status and the NIO Fault Status 1 register for errors.
6. Check data in CM for errors.

This subsection consists of 13 conditions with each condition being a different pattern.

```
( optional error data )  TEST SEED ssss
                         XFER BLOCK cccc bbbb tttt tttt
```

Subsection 3

Tests the assembly buffer ranks.

Method of Execution:

1. Bypass this subsection if PARAM15 is set.
2. Generate a data pattern in the PP.
3. Block CM requests CMI.
4. Load the T register is with the CM address and byte count of 16 (two CM words).
5. Initiate a PP to CM data transfer.
6. Output 64 PP words by the PP. CM requests are then allowed to the CMI allowing the transfer to complete.
7. Check the DMA PP/IPI mode and T' register Empty status bits to ensure that they are set in the Operational Status register.
8. Check the NIO Fault Status 1 and Error Status registers for errors.
9. Check data in CM for errors.

This subsection consists of 13 conditions with each condition being a different pattern.

```
( optional error data )  TEST SEED ssss
                         XFER BLOCK cccc bbbb tttt tttt
```

Subsection 4

Tests the assembly buffer parity generators.

Method of Execution:

1. Generate a test pattern in the PP.
2. Load the T register with the CM address and byte count of 64 (eight CM words).
3. Initiate a PP to CM data transfer.
4. Output 256 PP words by the PP.
5. Check the NIO Fault Status 1 and Error Status registers for errors.
6. Check data in CM for errors.

This subsection consists of four conditions with each condition being a different data pattern.

```
( optional error data )  TEST SEED ssss
                         XFER BLOCK cccc bbbb tttt tttt
```

Subsection 5

Tests the assembly buffer lower byte parity checker.

Method of Execution:

1. Generate a data pattern in the PP.
2. Enable Force Channel Input Parity 1 Low (force error code 11_{16}) in the IPI adapter Control register.
3. Load the T register with the CM address and byte count of eight (one CM word).
4. Initiate a PP to CM data transfer.
5. Output four PP words by the PP.
6. Check the DMA Error register status and the NIO Fault Status 1 register for errors.
7. Check data in CM for errors.

This subsection consists of three conditions with each condition being a different data pattern.

```
( optional error data )  TEST SEED ssss
                         XFER BLOCK cccc bbbb tttt tttt
```

Subsection 6

Tests the assembly buffer upper byte parity checker.

Method of Execution:

Same as subsection 4 only Force Channel Input Parity Bit 0 Low (force error code 10_{16}) is used.

```
( optional error data )  TEST SEED ssss
                         XFER BLOCK cccc bbbb tttt tttt
```

Section 06 – IPI Static Force Error Code Function Test

This section tests the IPI static force error code functions. The section uses the IPI Error Register status and the DMA Error register status (IPI error bit 55) to test for the forced errors.

Subsection 0

Tests the response on Read IPI Error register function.

Method of Execution:

1. Issue a Clear IPI Error (0022_{16}) function.
2. Issue a Read IPI Error register ($00F1_{16}$) function to the IPI adapter.
3. Read and check the IPI Error register to be zero.
4. Check the DMA Error register status and CIO Fault Status 2 register for errors.

```
( optional error data ) none.
```

Subsection 1

Tests the IPI error forcing functions that statically produce errors (that is, those error codes that do not require data transfers or other actions).

Method of Execution:

1. Issue one of the IPI error forcing functions listed below to the IPI adapter.
2. Read and check the IPI Error register and DMA Error register status for the appropriate response.
3. Check the CIO Channel Error Flag and CIO Fault Status 2 register for errors.

The subsection also tests that Master Clear Adapter (0000_{16}) function and Clear Error (0100_{16}) function clear the IPI Error register and DMA Error Status register for each error forced.

4. The MAC IPI adapter Status register for IPI error in MAC Byte 0 and the appropriate MAC Byte 2 status as listed below.

This subsection consists of seven conditions, each condition being a different force error function.

```
( optional error data )  FORCE ERROR FUNCTION= FFFF

BUFFER COUNTER 1 PARITY (0522(16)) (MAC byte 2, BUFFER COUNTER PE)
BUFFER COUNTER 2 PARITY (0622(16)) (MAC byte 2, BUFFER COUNTER PE)
BUFFER WRITE ADDRESS PARITY (0722(16)) (MAC byte 2, BUFFER COUNTER PE)
BUFFER READ ADDRESS 1 PARITY (0822(16)) (MAC byte 2, BUFFER COUNTER PE)
BUFFER READ ADDRESS 2 PARITY (0922(16)) (MAC byte 2, BUFFER COUNTER PE)
SYNC COUNTER PARITY (0A22(16)) (MAC byte 2, SYNC COUNTER PE)
PERIOD COUNTER PARITY (0B22(16)) (MAC byte 2, PERIOD COUNTER PE)
```

Subsection 2

Tests the response on a Read IPI Status register function.

Method of Execution:

1. Issue a Clear IPI Error (0022_{16}) function.
2. Issue a Read IPI Status register ($00E1_{16}$) function to the IPI adapter.
3. Read and check the IPI Status register to be zero.
4. Check the DMA Error register status and CIO Fault Status 2 register for errors.

```
( optional error data ) none.
```

Subsection 3

Tests the Clear IPI Error Interface Error register function.

Method of Execution:

1. Issue one of the IPI error forcing functions, listed in Subsection 1 of this section, to the IPI adapter.
2. Read and check the IPI Error register status for the appropriate response.
3. Issue the Clear IPI Interface Error register function to the IPI adapter.
4. Read and check the IPI Error register and DMA Error register status for the appropriate response.
5. Check the CIO Fault Status 2 register for errors.

This subsection consists of seven conditions, each condition being a different force error function.

```
( optional error data )  FORCE ERROR FUNCTION= FFFF
```

Subsection 4

Tests if the Master Clear clears the IPI Interface Error register.

Method of Execution:

1. Issue one of the IPI error forcing functions, listed in Subsection 1 of this section, to the IPI adapter.
2. Read and check the IPI Error register status for the appropriate response.
3. Issue the Master Clear function to the IPI adapter.
4. Read and check the IPI Error register and DMA Error register status for the appropriate response.
5. Check the CIO Fault Status 2 register for errors.

This subsection consists of seven conditions, each condition being a different force error function.

```
( optional error data )  FORCE ERROR FUNCTION= FFFF
```

Subsection 5

Tests the response on invert input data parity to BAS.

Method of Execution:

1. Set the Invert Input Data Parity to BAS (force error code 08_{16}) function in the Control register force error code.
2. Read and check the IPI Status register and DMA Error register status for errors.

```
( optional error data ) none.
```

Subsection 6

Tests the response on force output parity error upper.

Method of Execution:

1. Set the Force Output Parity Error Upper (force error code $0C_{16}$) function in the Control register force error code.
2. Output and input a data word of zeros on the IPI adapter CIO channel.
3. Read and check the Operational register status, IPI Status register, and DMA Error register status for errors.

```
( optional error data ) none.
```

Subsection 7

Tests the response on force output parity error lower.

Method of Execution:

1. Set the Force Output Parity Error Lower (force error code $0D_{16}$) function in the Control register force error code.
2. Output and input a data word of zeros on the IPI adapter CIO channel.
3. Read and check the Operational register status, IPI Status register, and DMA Error register status for errors.

```
( optional error data ) none.
```

Section 07 – IPI Channel Clock Select Function Test

This section tests the IPI adapter response to the functions for Attention Present ($00C1_{16}$) and Set Sync Period ($XX42_{16}$) on the IPI channel and External Clock/Port Select ($0X62_{16}$) on the IPI channel.

Subsection 0

Tests the IPI adapter response to the Attention Present function.

Method of Execution:

1. Issue the Master Clear (0000_{16}) function, to clear the IPI adapter.
2. Issue the Attention Present ($00C1_{16}$) function to the IPI adapter.
3. Check the CIO channel for being inactive.

   No attempt is made to read the IPI status that should be returned as a result of issuing this function to the IPI adapter.
4. Check the DMA Error register status and IPI Error register status for errors.

   (optional error data) none.

Subsection 1

Tests the IPI adapter response to the Set Sync Period on the IPI channel functions.

Method of Execution:

1. Issue the Master Clear (0000_{16}) function, to clear the IPI adapter.
2. Issue one of the Set Sync Period ($XX42_{16}$) functions listed below to the IPI adapter.
3. Check the CIO channel for being inactive.
4. Check the DMA Error register status and IPI Error register status for errors.

This subsection consists of 26 conditions, each being a different Set Sync Period function.

Function (Hex)	20-MHz Internal Clock
0042	0.16 Mbyte
6842	0.80 Mbyte
6C42	1 Mbyte
7442	1.70 Mbyte
7642	2 Mbyte
7942	3 Mbyte
7A42	4 Mbyte
7C42	5 Mbyte
7D42	6 Mbyte
7E42	10 Mbyte

Function (Hex)	20-MHz Ext. Clk	12-MHz Ext. Clk
8042	0.16 Mbyte	0.09 Mbyte
E742	0.80 Mbyte	0.48 Mbyte
E842	0.83 Mbyte	0.50 Mbyte
EC42	1 Mbyte	0.60 Mbyte
F342	1.50 Mbyte	0.92 Mbyte
F442	1.70 Mbyte	1 Mbyte
F542	1.80 Mbyte	1.10 Mbyte
F642	2 Mbyte	1.20 Mbyte
F742	2.50 Mbyte	1.30 Mbyte
F842	2.90 Mbyte	1.50 Mbyte
F942	3 Mbyte	1.70 Mbyte
FA42	3.30 Mbyte	2 Mbyte
FB42	4 Mbyte	2.40 Mbyte
FC42	5 Mbyte	3 Mbyte
FD42	6.70 Mbyte	4 Mbyte
FE42	10 Mbyte	6 Mbyte

```
( optional error data )  FUNCTION CODE=FFFF
```

Subsection 2

Tests the IPI adapter response to the External Clock/Port Select on the IPI channel functions.

Method of Execution:

1. Issue the Master Clear (0000_{16}) function to clear the IPI adapter.
2. Issue one of the Set External Clock ($XX62_{16}$) functions listed below to the IPI adapter.
3. Check the CIO channel for being inactive.
4. Check, for only the functions that have a clock defined, the DMA Error register status and IPI Error register status for errors.

This subsection consists of 16 conditions, each being a different Set External Clock function.

Function (Hex)	Clock	Port
0062	20 MHz	A
0162	12 MHz	A
0262	TBD	A
0362	TBD	A
0462	TBD	A
0562	TBD	A
0662	TBD	A
0762	TBD	A
0862	20 MHz	B
0962	12 MHz	B
0A62	TBD	B
0B62	TBD	B
0C62	TBD	B
0D62	TBD	B
0E62	TBD	B
0F62	TBD	B

```
( optional error data )  FUNCTION CODE=FFFF
```

Section 08 - IPI Bus Status And Control Test

This section tests the IPI Bus Control functions, Bus B Status, and IPI Status register. The section uses the Bus Control ($XXX1_{16}$) function, its Bus B Status response and the Read IPI Status register ($00E0_{16}$) function to check the operation of the IPI control lines, Select Out, Master Out, and Sync Out.

Subsection 0

Tests the IPI master reset sequence and Sync Out signal.

Method of Execution:

1. Issue the Bus Control function (0013_{16}).
2. Set Sync Out, clear Master Out, clear Select Out, and set condition code 1, is issued and the CIO channel activated in order to read the IPI Bus B Status word.
3. Check the IPI Bus B status and IPI Status register for Sync Out set and all other bits clear.
4. Check the MAC IPI adapter Status register's IPI Status register Byte 7 for Sync Out status.
5. Issue, at least 12 microseconds later, the Bus Control (0011_{16}) function with all control bits clear and Condition code 1 set.
6. Check the IPI Bus B status and IPI Status register for all bits clear.
7. Check the IPI Error register status and DMA Error register status for errors.

   (optional error data) none.

Subsection 1

Tests the select/deselect Bus Slave sequence.

Method of Execution:

1. Execute the Master Reset sequence with condition code 0.
2. Issue the Bus Control ($XX29_{16}$) function with Select Out and condition code 2 set, in Test mode and the CIO channel activated in order to read the IPI Bus B status word.
3. Check the IPI Bus B status and IPI Status register for Select Out and Slave In set and all other bits clear.
4. Check the MAC IPI adapter Status register's IPI Status register Byte 7 for Select Out and Slave In status.
5. Issue the Bus Control ($XX71_{16}$) function with Select Out clear and condition code 7 set.
6. Check IPI Bus B status and IPI Status register for Select Out and Slave In cleared.
7. Check the IPI Error register status and DMA Error register status for errors.

   (optional error data) none.

Subsection 2

Tests the Master Out signal.

Method of Execution:

1. Execute the Master Reset sequence with condition code 0.
2. Enable Test mode in the IPI adapter Control register.
3. Issue the Bus Control ($XX15_{16}$) function with Master Out and condition code 1 set.
4. Activate the CIO channel in order to read the IPI Bus B status word.
5. Check the IPI Bus B status and IPI Status register for Master Out set and all other bits clear.
6. Check the MAC IPI adapter Status register's IPI Status register Byte 7 for Master Out status.
7. Issue the Bus Control ($XX11_{16}$) function with Select Out clear and condition code 1 set.
8. Check the IPI Bus B status and IPI Status register for Select Out clear.
9. Check the IPI Error register status and DMA Error register status for errors.

   (optional error data) none.

Subsection 3

Tests the Sync Out signal.

Method of Execution:

1. Execute the Master Reset sequence with condition code 0.
2. Enable Test mode in the IPI adapter Control register.
3. Issue the Bus Control ($XX29_{16}$) function with Select Out and condition code 2 set.
4. Activate the CIO channel in order to read the IPI Bus B status word.
5. Check the IPI Bus B status and IPI Status register for Select Out and Slave In set with all other bits clear.
6. Check the MAC IPI adapter Status register's IPI Status register Byte 7 for Select Out and Slave In status.
7. Issue the Bus Control ($XX5B_{16}$) function with Select Out and Sync Out with condition code 5 set.
8. Activate the CIO channel in order to read the IPI Bus B status word.
9. Check the IPI Bus B status and IPI Status register for Select Out, Sync In, and Sync Out set with all other bits clear.
10. Check the MAC IPI adapter Status register's IPI Status register Byte 7 for Select Out, Sync Out, and Slave In status.
11. Issue the Bus Control ($XX69_{16}$) function with Select Out with condition code 6 set.
12. Activate the CIO channel in order to read the IPI Bus B status word.
13. Check the IPI Bus B status and IPI Status register for Select Out set with all other bits clear.
14. Check the MAC IPI adapter Status register's IPI Status register Byte 7 for Select Out status.
15. Issue the Bus Control ($XX01_{16}$) function with Select Out, Master Out, and Sync Out clear with condition code 0 set.
16. Check the IPI Error register status and DMA Error register status for errors.

    (optional error data) none.

Subsection 4

Tests the IPI illegal operation.

Method of Execution:

1. Issue the Bus Control (0041_{16}) function with Sync Out clear, Master Out clear, Select Out clear, and condition code 41 set).
2. Activate the CIO channel in order to read the IPI Bus B status word.
3. Check the IPI Bus B status and IPI Status register for the error bit set with all other bits clear.
4. Check the MAC IPI adapter Status register's IPI Status register Byte 7 for the error bit set.
5. Check the IPI Error register status ($00F1_{16}$) for the illegal operation bit set.
6. Check the DMA Error register status for the IPI error and LZ error bits set.
7. Issue the Bus Control ($XX01_{16}$) function with Select Out, Master Out, and Sync Out clear with condition code 0 set).
8. Check the IPI Status register for all bits clear.
9. Check the MAC IPI adapter Status register's IPI Status register Byte 7 for all bits clear.
10. Check the IPI Error register status ($00F1_{16}$) for all bits clear.
11. Check the DMA Error register status for all bits clear.

(optional error data) none.

Subsection 5

Tests the Force ICI Output Data Parity Lower and Upper.

Method of Execution:

1. Execute the Master Reset sequence with condition code 0.
2. Issue the Force ICI Output Data Parity Lower ($0C22_{16}$) function to the IPI adapter.
3. Enable Test mode in the IPI adapter Control register.
4. Issue the Bus Control ($XX29_{16}$) function with Select Out and condition code 2 set.
5. Check the IPI Bus B status for Select Out and Slave In set and all other bits clear.
6. Check the IPI Status register for Error, Select Out and Slave In set and all other bits clear.
7. Check the IPI Error register status and DMA Error register status for errors.
8. Check the DMA Error register status for IPI error and LZ error bits set.
9. Issue a Clear IPI Error (0022_{16}) function to the IPI adapter.
10. Check the DMA Error register status for IPI error and LZ error bits set.
11. Issue the Bus Control ($XX01_{16}$) function with Select Out, Master Out, and Sync Out clear with condition code 0 set.
12. Check the IPI Error register status and DMA Error register status for errors.

The subsection is repeated for Force ICI Output Parity Upper ($0D22_{16}$).

```
( optional error data ) none.
```

Subsection 6

Tests the Force ICI Input Data Parity Lower and Upper.

Method of Execution:

1. Execute the Master Reset sequence with condition code 0.
2. Issue the Force ICI Input Data Parity Lower ($0E22_{16}$) function to the IPI adapter.
3. Enable Test mode in the IPI adapter Control register.
4. Issue the Bus Control ($XX29_{16}$) function with Select Out and condition code 2 set.
5. Check the IPI Bus B status for Error, Select Out, and Slave In set and all other bits clear.
6. Check the IPI Error register status and DMA Error register status for errors.
7. Check the IPI Error register status for errors.
8. Issue a Clear IPI Error (0022_{16}) function to the IPI adapter.
9. Check the DMA Error register status for IPI error and LZ error bits set.
10. Issue the Bus Control ($XX01_{16}$) function with Select Out, Master Out, and Sync Out clear with condition code 0 set.
11. Check the IPI Error register status and DMA Error register status for errors.

    The subsection is repeated for force ICI input parity upper ($0F22_{16}$).

    (optional error data) none.

Subsection 7

Tests the IPI illegal sequence status bit.

Method of Execution:

1. Execute the Master Reset sequence with condition code 0.
2. Enable Test mode in the IPI adapter Control register.
3. Issue the Bus Control ($XX15_{16}$) function with Master Out and condition code 1 set.
4. Check the IPI Bus B status for Master Out set and all other bits clear.
5. Check the IPI Error register status for errors.
6. Issue the Bus Control ($XX2D_{16}$) function with Select Out, Master Out, and condition code 2 set.
7. Check the IPI Bus B status for Error, Select Out, Master Out and Slave In set and all other bits clear.
8. Check the IPI Error register status and DMA Error register status for errors.
9. Check the IPI Error register status for errors.

   (optional error data) none.

Section 09 – IPI Adapter Random Vector Generator Test

This section tests the IPI adapter Random Vector Generator. The section uses the Read Operand Generator (0004_{16}) and Write Operand Generator (0014_{16}) functions and the IPI adapters Test mode facility in PP Input mode.

Subsection 0

Tests the Random Vector Generator Test Seed Register and interface to the Channel Data Register.

Method of Execution:

1. Load data into the Random Vector Generator using the Write Operand Generator (0014_{16}) function.
2. Read the Random Vector Generator using the Read Operand Generator (0004_{16}) function.
3. Check the CIO Channel Error Flag for errors.
4. Check data for errors.

This subsection consists of 20 conditions each being a different data pattern.

```
( optional error data )  TEST SEED ssss
```

Subsection 1

Tests the Random Vector Generator to Channel Data register interface using the IPI input buffer.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select the Random Vector Generator and initialize with a data pattern.
4. Place the IPI adapter in Test mode.
5. Initiate an IPI channel to PP data transfer in 16-bit Interlocked mode.
6. Input one data word from the IPI adapter.
7. Check, after completion of the data transfer, the IPI Error register status and DMA Error register status for errors.
8. Check the NIO Fault Status 1 register for errors.
9. Check data input from the IPI adapter for errors.

This subsection consists of 20 conditions each being a different data pattern.

Subsection 2

Tests the Random Vector Generator to Channel register via block transfers.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select and intitialize the Random Vector Generator with a random data pattern.
4. Place the IPI adapter in Test mode.
5. Initiate an IPI channel to PP data transfer in 16-bit Interlocked mode.
6. Input 257 words from the IPI adapter.
7. Check, after completion of the data transfer, the IPI Error register status and DMA Error register status for errors.
8. Check data input from the IPI adapter for errors.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 3

Test Force Lost Data Error status.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select and initialize the Random Vector Generator with a PP unique data pattern.
4. Place the IPI adapter in Test mode.
5. Initiate an IPI channel to PP data transfer in 16-bit Interlocked mode.
6. Input 256 words from the IPI adapter with one word remaining in the IPI adapter.
7. Check, after completion of the data transfer, the IPI Error register status and DMA Error register status for errors.
8. Check data input from the IPI adapter for errors.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 4

Test Force Buffer Not Empty status.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select and initialize the Random Vector Generator with a PP unique data pattern.
4. Place the IPI adapter in Test mode.
5. Initiate an IPI channel to PP data transfer in 16-bit Interlocked mode.
6. Input 254 words from the IPI adapter with three words remaining in the IPI adapter.
7. Check, after completion of the data transfer, the IPI Status register status, IPI Error register status and DMA Error register status for errors.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Section 10 – IPI Channel Adapter Compare Unit Test

This section tests the IPI adapter Compare unit using the Random Vector Generator Test mode facility in PP Output mode.

Subsection 0

Tests the IPI adapter Compare unit.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select and initialize the Random Vector Generator with a data pattern.
4. Place the IPI adapter in Test mode.
5. Initiate a PP to IPI channel data transfer in 16-bit Interlocked mode.
6. Place a data word in the PPBUF data buffer.
7. Output one data word to the IPI adapter.
8. Check, after completion of the data transfer, the IPI Error register status and DMA Error register status for errors.
9. Read and compare the test seed with the expected result.

This subsection consists of 19 conditions each being a different pair of operands.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 1

Tests the IPI adapter Compare unit in Block mode.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select and initialize the Random Vector Generator with a data pattern.
4. Place the IPI adapter in Test mode.
5. Initiate a PP to IPI channel data transfer in 16-bit Interlocked mode.
6. Output 19 data words to the IPI adapter.
7. Check, after completion of the data transfer, the IPI Error register status and DMA Error register status for errors.
8. Read and compare the test seed with the expected result.

This subsection consists of 19 conditions each being a different test seed value.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Section 11 – Deadman Timeout Test

This section tests the deadman timer, the disable timeout control, and timeout status.

Subsection 0

Test deadman timeout.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Select and initialize the Random Vector Generator with a data pattern.
3. Place the IPI adapter in Test mode.
4. Initiate a PP to IPI channel data transfer in 16-bit Interlocked mode.
5. Output one data word on the IPI adapter; the program waits for 1.2 seconds for the deadman timer to expire and set the CIO Channel Error Flag.
6. Check the MAC IPI adapter Status register's DMA Error register status Byte 1 for Timeout status.
7. Check the NIO Fault Status 2 register for errors.
8. Check the IPI Error register status and the DMA Error register status for errors.

   (optional error data) none.

Subsection 1

Test inhibit deadman timeout.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Select and initialize the Random Vector Generator with a data pattern.
3. Place the IPI adapter in Test mode with disable IPI timeout enabled.
4. Initiate a PP to IPI channel data transfer in 16-bit Interlocked mode.
5. Output one data word on the IPI adapter; the program waits for 1.2 seconds for the deadman timer to expire and set the CIO Channel Error Flag.
6. Check the MAC IPI adapter Status register's DMA Error register status Byte 1 for errors.
7. Check the NIO Fault Status 2 register for errors.
8. Check the IPI Error register status and the DMA Error register status for errors.

   (optional error data) none.

Section 12 – PP Transfer Test (16-Bit Mode)

This section is the PP transfer test. Data is transferred between the PP and the Random Vector Generator in the IPI channel and between the IPI channel and PP in 16-bit mode.

Subsection 0

Tests IPI/PP transfers in 16-bit Interlocked mode.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select and initialize the Random Vector Generator with a random data pattern.
4. Place the IPI adapter in Test mode.
5. Initiate an IPI channel to PP data transfer in 16-bit Interlocked mode.
6. Input 200 words from the IPI adapter.
7. Check, after completion of the data transfer, the IPI Error register status and DMA Error register status for errors.
8. Check the data input from the IPI adapter for errors.

This subsection consists of 10 conditions each being a different random data pattern.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 1

Tests PP/IPI transfers in 16-bit Interlocked mode.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select and initialize the Random Vector Generator with a random data pattern.
4. Place the IPI adapter in Test mode.
5. Initiate an IPI channel to PP data transfer in 16-bit Interlocked mode.
6. Output 200 words to the IPI adapter.
7. Check, after completion of the data transfer, the IPI Error register status and DMA Error register status for errors.
8. Read and compare the test seed with the expected result.

This subsection consists of 10 conditions each being a different random data pattern.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 2

Tests IPI/PP transfers in 16-bit Streaming mode.

Method of Execution:

Same as subsection 0, only 16-bit Streaming mode is used.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 3

Tests PP/IPI transfers in 16-bit Streaming mode.

Method of Execution:

Same as subsection 1, only 16-bit Streaming mode is used.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 4

Tests Force IPI Bus A and B Output Parity.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select and initialize the Random Vector Generator with a random data pattern.
4. Issue the Force IPI Bus A Output Parity (0122_{16}) function to the IPI adapter.
5. Place the IPI adapter in Test mode.
6. Initiate a PP to IPI channel data transfer in 16-bit Interlocked mode.
7. Output 200 words to the IPI adapter.
8. Disable, after completion of the data transfer, Test mode.
9. Check the IPI IPI Error register status and DMA Error register status for errors.

The subsection is repeated for the Force IPI Bus B Output Parity (0222_{16}) function.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 5

Tests Force IPI Bus A and B Input Parity.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select and initialize the Random Vector Generator with a random data pattern.
4. Issue the Force IPI Bus A Input Parity (0322_{16}) function to the IPI adapter.
5. Place the IPI adapter in Test mode.
6. Initiate a PP to IPI channel data transfer in 16-bit Interlocked mode.
7. Output 200 words to the IPI adapter.
8. Disable, after completion of the data transfer, Test mode.
9. Check the IPI IPI Error Register status and DMA Error Register status for errors.

The subsection is repeated for the Force IPI Bus B Input Parity (0422_{16}) function.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Section 13 – PP Transfer Test (8-Bit Mode)

This section is the PP transfer test. That is, data is transferred between the PP and the Random Vector Generator in the IPI channel and between the IPI channel and PP in 8-bit mode.

Subsection 0

Tests IPI/PP transfers in 8-bit Interlocked mode.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select and initialize the Random Vector Generator with a random data pattern.
4. Place the IPI adapter in Test mode.
5. Initiate an IPI channel to PP data transfer in 8-bit Interlocked mode.
6. Input 200 words from the IPI adapter.
7. Check, after completion of the data transfer, the IPI Error register status and DMA Error register status for errors.
8. Check the data input from the IPI adapter for errors.

This subsection consists of 10 conditions each being a different random data pattern.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 1

Tests PP/IPI transfers in 8-bit Interlocked mode.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select and initialize the Random Vector Generator with a random data pattern.
4. Place the IPI adapter in Test mode.
5. Initiate an IPI channel to PP data transfer in 8-bit Interlocked mode.
6. Output 200 words to the IPI adapter.
7. Check, after completion of the data transfer, the IPI Error register status and DMA Error register status for errors.
8. Read and compare the test seed with the expected result.

This subsection consists of 10 conditions each being a different random data pattern.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 2

Tests IPI/PP transfers in 8-bit Streaming mode.

Method of Execution:

Same as subsection 0, only 8-bit Streaming mode is used.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 3

Tests PP/IPI transfers in 8-bit Streaming mode.

Method of Execution:

Same as subsection 1, only 8-bit Streaming mode is used.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Section 14 – Transfer Complete Test

This section tests setting of transfer complete at the end of a PP mode transfer.

Subsection 0

Test transfer complete does not set the CIO Channel flag in PP mode.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Select and initialize the Random Vector Generator with a PP unique data pattern.
3. Place the IPI adapter in Test mode with Enable TIP Flag set in the IPI adapter Control register.
4. Initiate a PP to IPI channel data transfer in 16-bit Interlocked mode.
5. Output one data word is to the IPI adapter.
6. Check, after completion of the data transfer, the IPI Error register status and DMA Error register status for errors.
7. Check the CIO channel flag for being clear.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Section 15 – Central to IPI Transfer Register Interface

This section tests the CM to IPI Transfer register interface using the Test mode facility in DMA Read mode and invert parity.

Subsection 0

Tests transfer complete setting of CIO channel flag.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select and initialize the Random Vector Generator with a PP unique data pattern.
4. Place the IPI adapter in Test mode with Enable TIP Flag set in the Control register.
5. Initiate a CM to IPI channel data transfer in 16-bit Interlocked mode using the DMA Write To IPI ($0D00_{16}$) function.
6. Load the T register with the CM address and byte count of eight (one CM word).
7. Check, after completion of the data transfer, the IPI Error register status and DMA Error register status for errors.
8. Check the CIO channel flag for being set.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 1

Tests the disassembly buffer ranks.

Method of Execution:

1. Bypass this subsection if PARAM15 is set.
2. Execute the IPI Master Reset sequence.
3. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
4. Select and initiate the Random Vector Generator with a data pattern.
5. Place the IPI adapter in Test mode with Enable TIP Flag set in the IPI adapter Control register.
6. Block CM requests at the CMI.
7. Initiate a CM to IPI channel data transfer in 16-bit Interlocked mode using the DMA Write To IPI ($0D00_{16}$) function.
8. Load the T register with the CM address and byte count of eight (one CM word).
9. Release CM requests at the CMI to allow the transfer to complete.
10. Check, after completion of the data transfer, the IPI Error register status and DMA Error register status for errors.
11. Read and check the test seed for errors.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 2

Tests the disassembly buffer.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select and initialize the Random Vector Generator with a data pattern.
4. Place the IPI adapter in Test mode with Enable TIP Flag set in the IPI adapter Control register.
5. Initiate a CM to IPI channel data transfer in 16-bit Interlocked mode using the DMA Write To IPI ($0D00_{16}$) function.
6. Load the T register with the CM address and byte count of eight (one CM word).
7. Check, after completion of the data transfer, the IPI Error register status, DMA Error register status and NIO Fault Status 1 register for errors.
8. Read and check the test seed for errors.

This subsection consists of 13 conditions each being a different data pattern. The real memory address and byte count remain unchanged for each condition.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 3

Tests the disassembly buffer parity generators.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select and initialize the Random Vector Generator with a data pattern.
4. Place the IPI adapter in Test mode with Enable TIP Flag set in the IPI adapter Control register.
5. Initiate a CM to IPI channel data transfer in 16-bit Interlocked mode using the DMA Write To IPI ($0D00_{16}$) function.
6. Load the T register with the CM address and byte count of eight (one CM word).

7. Check, after completion of the data transfer, the IPI Error register status, DMA Error register status and NIO Fault Status 1 register for errors.

8. Read and check the test seed for errors.

   This subsection consists of 32 conditions each being a different data pattern. The real memory address and byte count remain unchanged for each condition.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 4

Tests the assembly buffer lower byte parity checker.

Method of Execution:

1. Execute the IPI Master Reset sequence.

2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.

3. Select and initialize the Random Vector Generator with a data pattern.

4. Place the IPI adapter in Test mode with Invert Lower Channel Output Parity (force error code 14_{16}) set in the Control register force error code and Enable TIP Flag set in the IPI adapter Control register.

5. Initiate a CM to IPI channel data transfer in 16-bit Interlocked mode using the DMA Write To IPI ($0D00_{16}$) function.

6. Load the T register with the CM address and byte count of eight (one CM word).

7. Check, after completion of the data transfer, the IPI Error register status, DMA Error register status and NIO Fault Status 1 register for errors.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 5

Tests the assembly buffer upper byte parity checker.

Method of Execution:

Same as subsection 4 only Invert Upper Channel Output Parity (force error code 13_{16}) is set in the Control register force error code.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Section 16 – IPI To Central Memory Interface Test

This section tests the IPI Transfer register to CM interface utilizing the Test mode facility in DMA Write mode and invert parity.

Subsection 0

Tests transfer complete setting of CIO channel flag.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select the Random Vector Generator with a PP unique data pattern.
4. Place the IPI adapter in Test mode with Enable TIP Flag set in the Control register.
5. Initiate an IPI channel to CM data transfer in 16-bit Interlocked mode using the DMA Read From IPI ($0C00_{16}$) function.
6. Load the T register with the CM address and byte count of eight (one CM word).
7. Check, after the transfer completes, the CIO Channel Flag and DMA Error register status for errors.

```
( optional error data )  TEST SEED ssss
                                XFER BLK bbbb tttt tttt
```

Subsection 1

Tests the IPI Transfer register to CM interface.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select and initiate the Random Vector Generator with a data pattern.
4. Place the IPI adapter in Test mode.
5. Initiate an IPI channel to CM data transfer in 16-bit Interlocked mode using the DMA Read From IPI ($0C00_{16}$) function.
6. Load the T register with the CM address and byte count of eight (one CM word).

7. Check, after the data transfer completes, the DMA Error Status Register and NIO Fault Status 1 Register for errors.

8. Check data in CM for errors.

This subsection consists of 13 conditions each being a different pattern. The real memory address and byte count remain unchanged for each condition.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 2

Tests the assembly buffer ranks.

Method of Execution:

1. Bypass this subsection PARAM15 is set.

2. Executethe IPI Master Reset sequence.

3. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.

4. Select and initialize the Random Vector Generator with a data pattern.

5. Place the IPI adapter in Test mode.

6. Block CM requests at the CMI.

7. Initiate an IPI channel to CM data transfer in 16-bit Interlocked mode using the DMA Read From IPI ($0C00_{16}$) function.

8. Load the T register with the CM address and byte count of eight (one CM word).

9. Release CM requests at the CMI to allow the transfer to complete.

10. Check, after the data transfer completes, the DMA Error Status register and NIO Fault Status 1 register for errors.

11. Check data in CM for errors.

This subsection consists of 13 conditions each being a different data pattern. The real memory address and byte count remain unchanged for each condition.

```
( optional error data )  TEST SEED ssss
                          XFER BLK bbbb tttt tttt
```

Subsection 3

Tests the assembly buffer parity generators.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select and initialize the Random Vector Generator with a data pattern.
4. Place the IPI adapter in Test mode.
5. Initiate an IPI channel to CM data transfer in 16-bit Interlocked mode using the DMA Read From IPI ($0C00_{16}$) function.
6. Load the T register with the CM address and byte count of eight (one CM word).
7. Check, after the data transfer completes, the DMA Error Status register and NIO Fault Status 1 register for errors.
8. Check data in CM for errors.

This subsection consists of four conditions each being a different data pattern. The real memory address and byte count remain unchanged for each condition.

```
( optional error data )  TEST SEED ssss
                          XFER BLK bbbb tttt tttt
```

Subsection 4

Tests the assembly buffer lower byte parity checker.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select and initialize the Random Vector Generator with a data pattern.
4. Place the IPI adapter in Test mode with Force Assembly Buffer Lower Byte Parity (force error code 11_{16}) set in the Control register force error code.
5. Initiate an IPI channel to CM data transfer in 16-bit Interlocked mode using the DMA Read From IPI ($0C00_{16}$) function.
6. Load the T register with the CM address and byte count of eight (one CM word).
7. Check, after the data transfer completes, CIO Channel Flag, the DMA Error Status register and NIO Fault Status 1 register for errors.

This subsection consists of three conditions each being a different data pattern. The real memory address and byte count remain unchanged for each condition.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 5

Tests the assembly buffer upper byte parity checker.

Same as subsection 4 only Force Assembly Buffer Upper Byte Parity (force error code 10_{16}) is set in the Control register force error code.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Section 17 – DMA Transfer Register Address Incrementer Test

This section tests the DMA Transfer register address incrementer utilizing the Test mode facility in DMA Read mode.

Subsection 0

Tests the DMA Transfer register address incrementer.

Method of Execution:

1. Initialize the CM address that will be read if the address is within the defined CM buffer boundaries.
2. Executethe IPI Master Reset sequence.
3. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
4. Select and initialize the Random Vector Generator with a PP unique data pattern.
5. Place the IPI adapter in Test mode.
6. Initiate a CM to IPI channel data transfer in 16-bit Interlocked mode using the DMA Write To IPI ($0D00_{16}$) function.
7. Load the T register with the CM address and byte count of eight (one CM word).
8. Check, after the data transfer completes, the DMA Error register status and CIO Fault Status 2 register for errors.
9. Read and check the T register for errors.

This subsection consists of 56 conditions each being a different CM address pattern.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 1

Tests force parity prediction error.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select and initialize the Random Vector Generator with a all ones data pattern.
4. Place the IPI adapter in Test mode with Force Address Parity Prediction Error (force error code 15_{16}) set in the Control register force error code.
5. Initiate an IPI channel to CM data transfer in 16-bit Interlocked mode using the DMA Read From IPI ($0C00_{16}$) function.
6. Load the T register with the CM address and byte count of eight (one CM word).
7. Check, after the data transfer completes, the DMA Error register status and NIO Fault Status 1 register for errors.
8. Check data in CM for errors.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Section 18 – Internal Clock Transfer Test

This section tests the data transfers using the internal clock in the IPI adapter.

Subsection 0

Tests Port A internal clock transfer test.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Select Port A. Set sync period to 20-MHz internal clock and set transfer rate to one of the transfer rates as determined by the number of the condition.
3. Select and initialize the Random Vector Generator with a random data pattern.
4. Place the IPI adapter in Test mode.
5. Initiate an IPI channel to CM data transfer in 16-bit Interlocked mode using the DMA Read From IPI ($0C00_{16}$) function.
6. Load the T register with the CM address and byte count of 128 (16 CM words).
7. Check, after the data transfer completes, the DMA Error register status for errors.
8. Execute the IPI Master Reset sequence.
9. Select Port A, set sync period to 20-MHz internal clock, and set the transfer rate to one of the transfer rates as determine by the number of the condition.
10. Place the IPI adapter in Test mode.
11. Initiate a CM to IPI channel data transfer in 16-bit Interlocked mode using the DMA Write To IPI ($0D00_{16}$) function.
12. Load the T register with the CM address and byte count of 128 (16 CM words).
13. Check, after the data transfer completes, the DMA Error register status for errors.
14. Check the vector in the Test mode operand generator for errors.

This subsection consists of 10 conditions each being a different random data pattern.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 1

Tests Port B internal clock transfer test.

Method of Execution:

Same as Subsection 0, but Port B is used instead of Port A.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Section 19 – Port A External Clock Test

This section tests the data transfers using the external clock and Port A in the IPI adapter.

Subsection 0

Tests transfers using the 24-MHz external clock on Port A.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Select Port A and the 24-MHz external clock and set the transfer rate to the transfer rate as determined by the number of the condition.
3. Select and initialize the Random Vector Generator with a random data pattern.
4. Place the IPI adapter in Test mode.
5. Initiate an IPI channel to CM data transfer in 16-Bit Interlocked mode using the DMA Read From IPI ($0C00_{16}$) function.
6. Load the T register with the CM address and byte count of 128 (16 CM words).
7. Check, after the data transfer completes, the DMA Error register status for errors.
8. Execute the IPI Master Reset sequence.
9. Select Port A and the 24-MHz external clock and set the transfer rate to the transfer rate as determined by the number of the condition.
10. Select and initialize the Random Vector Generator with a random data pattern.
11. Place the IPI adapter in Test mode.
12. Initiate a CM to IPI channel data transfer in 16-bit Interlocked mode using the DMA Write To IPI ($0D00_{16}$) function.
13. Load the T register with the CM address and byte count of 128 (16 CM words).
14. Check, after the data transfer completes, the DMA Error register status for errors.
15. Read and check the vector in the Test mode operand generator for errors.

This subsection consists of 16 conditions each being a different transfer rate.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 1

Tests transfers using the 20-MHz external clock on Port A.

Method of Execution:

Same as Subsection 0, but the 20-MHz external clock on Port A is used.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 2

Tests transfers using the 12-MHz external clock on Port A.

Method of Execution:

Same as Subsection 0, but the 12-MHz external clock on Port A is used.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Section 20 – Port B External Clock Test

This section tests the data transfers using the external clock and Port B in the IPI adapter.

Subsection 0

Tests transfers using the 24-MHz external clock on Port B.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Select Port B and the 24-MHz external clock and set the transfer rate to the transfer rate as determined by the number of the condition.
3. Select and initiate the Random Vector Generator with a random data pattern.
4. Place the IPI adapter in Test mode.
5. Initiate an IPI channel to CM data transfer in 16-bit Interlocked mode using the DMA Read From IPI ($0C00_{16}$) function.
6. Load the T register with the CM address and byte count of 128 (16 CM words).
7. Check, after the data transfer completes, the DMA Error register status is checked for errors.
8. Execute the IPI Master Reset sequence.
9. Select Port B and the 24-MHz external clock and set the transfer rate to the transfer rate as determined by the number of the condition.
10. Select and initialize the Random Vector Generator with a random data pattern.
11. Place the IPI adapter in Test mode.
12. Initiate a CM to IPI channel data transfer in 16-bit Interlocked mode using the DMA Write To IPI ($0D00_{16}$) function.
13. Load the T register with the CM address and byte count of 128 (16 CM words).
14. Check, after the data transfer completes, the DMA Error register status for errors.
15. Read and check the vector in the Test mode operand generator for errors.

This subsection consists of 16 conditions each being a different transfer rate.

```
( optional error data )  TEST SEED ssss
                          XFER BLK bbbb tttt tttt
```

Subsection 1

Tests transfers using the 20-MHz external clock on Port B.

Method of Execution:

Same as Subsection 0, but the 20-MHz external clock on Port B is used.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 2

Tests transfers using the 12-MHz external clock on Port B.

Method of Execution:

Same as Subsection 0, but the 12-MHz external clock on Port B is used.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Section 21 – DMA Transfer Register Byte Count Test

This section tests the DMA Transfer register byte counter decrementer utilizing the Test mode facility in DMA Write mode.

Subsection 0

Tests the DMA Transfer register byte counter decrementer.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select and initialize the Random Vector Generator with a PP unique data pattern.
4. Place the IPI adapter in Test mode.
5. Initiate an IPI channel to CM data transfer in 16-bit Interlocked mode using the DMA Read From IPI ($0C00_{16}$) function.
6. Load the T register with the CM address and byte count for the maximum number of CM words available.
7. Check, after the data transfer completes, the DMA Error register status and NIO Fault Status 1 register for errors.
8. Check data in CM for errors.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Section 22 – DMA Transfer Error Test (Adapter Forced)

This section is considered the DMA transfer error test. Errors are forced in this section by utilizing the IPI adapters hardware functions.

Subsection 0

Tests response on CM response code parity error.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select and initialize the Random Vector Generator with a PP unique data pattern.
4. Place the IPI adapter in Test mode with Invert Central Memory Response Code Parity (force error code 05_{16}) set in the Control register force error code.
5. Initiate an IPI channel to CM data transfer in 16-bit Interlocked mode using the DMA Read From IPI ($0C00_{16}$) function.
6. Load the T register with the CM address and byte count of eight (one CM word).
7. Check, after the data transfer completes, the DMA Error Status register, CIO Channel Status register, NIO Fault Status 1 register, and CIO Fault Status 2 register for errors.
8. Check data in CM for errors.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 1

Tests response on invalid response code.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select and initialize the Random Vector Generator with a PP unique data pattern.
4. Place the IPI adapter in Test mode with Force Invalid Response Code (force error code 04_{16}) set in the Control register force error code.
5. Initiate an IPI channel to CM data transfer in 16-bit Interlocked mode using the DMA Read From IPI ($0C00_{16}$) function.
6. Load the T register with the CM address and byte count of eight (one CM word).
7. Check, after the data transfer completes, the DMA Error Status register, CIO Channel Status register, NIO Fault Status 1 register, and CIO Fault Status 2 register for errors.
8. Check data in CM for errors.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 2

Tests response on force byte count zero.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select and initialize the Random Vector Generator with a PP unique data pattern.
4. Place the IPI adapter in Test mode with Force Byte Count Zero (force error code 16_{16}) set in the Control register force error code.
5. Initiate a CM to IPI channel data transfer in 16-bit Interlocked mode using the DMA Write To IPI ($0D00_{16}$)) function.
6. Load the T register with the CM address and byte count of eight (one CM word).
7. Check, after the data transfer completes, the DMA Error Status register, CIO Channel Status register, NIO Fault Status 1 register, and CIO Fault Status 2 register for errors.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 3

Tests the false start transfer, byte count zero.

Method of Execution:

1. Initialize the CM buffer with the PP unique data pattern.
2. Execute the IPI Master Reset sequence.
3. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
4. Select and initialize the Random Vector Generator with a PP unique data pattern.
5. Place the IPI adapter in Test mode.
6. Initiate an IPI channel to CM data transfer in 16-bit Interlocked mode using the DMA Read From IPI ($0C00_{16}$) function.
7. Load the T register with the CM address and byte count of zero.
8. Check, after the data transfer completes, the DMA Error Status register, CIO Channel Status register, NIO Fault Status 1 register, and CIO Fault Status 2 register for errors.
9. Check data in CM for errors.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Section 23 – DMA Transfer Error Test (MR Forced)

This section is the DMA transfer error test. Errors are forced in this section by utilizing the I4 Test mode Maintenance register hardware functions. This section is bypassed if PARAM15 is set.

NOTE

Subsections 0, 3, 4, 5, and 6 are not executed when in concurrent maintenance mode.

Subsection 0

Test response on uncorrected write error.

Method of Execution:

1. Generate a data pattern in the PP.
2. Enable the maintenance channel's Force Data-Out Parity Bit Low (I4 Test mode Maintenance register test code 37B) at the CMI.
3. Load the T register with the CM address and byte count of eight (one CM word).
4. Initiate a PP to CM data transfer.
5. Output four PP words by the PP.
6. Check the DMA Error register status, NIO Fault Status 1 register and CIO Fault Status 2 register for errors.

This subsection consists of 12 conditions with each condition being a different pattern.

```
( optional error data )  MR TEST MODE =mm
                         TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 1

Test response on uncorrected read error.

Method of Execution:

1. Initialize the CM buffer with a PP unique data pattern.
2. Execute the IPI Master Reset sequence.
3. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
4. Select and initialize the Random Vector Generator with a data pattern.
5. Place the IPI adapter in Test mode.
6. Enable the maintenance channel's Force Data-In Parity Bit Low (I4 Test mode Maintenance register test code 36B) at the CMI.
7. Initiate a CM to IPI channel data transfer in 16-bit Interlocked mode using the DMA Write to IPI ($0D00_{16}$) function.
8. Load the T register with the CM address and byte count of eight (one CM word).
9. Check, after the data transfer completes, the DMA Error Status register, CIO Channel Status register, NIO Fault Status 1 register, and CIO Fault Status 2 register for errors.
10. Disable Force Data-In Parity Bit Low at the CMI.

    This subsection consists of eight conditions with each condition being a different pattern.

```
( optional error data )  MR TEST MODE =mm
                          TEST SEED ssss
                          XFER BLK bbbb tttt tttt
```

Subsection 2

Tests the response on CMI response code parity error.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select and initialize the Random Vector Generator with a PP unique data pattern.
4. Place the IPI adapter in Test mode.
5. Enable the maintenance channel's Force Response Code Parity Error (I4 Test mode Maintenance register test code 33B) at the CMI.
6. Initiate an IPI channel to CM data transfer in 16-bit Interlocked mode using the DMA Read To IPI ($0C00_{16}$) function.
7. Load the T register with the CM address and byte count of eight (one CM word).

8. Check, after the data transfer completes, the DMA Error Status register, CIO Channel Status register, NIO Fault Status 1 register, and CIO Fault Status 2 register for errors.

9. Disable the Force Response Code Parity Error at the CMI.

```
( optional error data )  MR TEST MODE =mm
                         TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 3

Tests the response on CMI reject.

Method of Execution:

1. Execute the IPI Master Reset sequence.

2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.

3. Select and initialize the Random Vector Generator with a PP unique data pattern.

4. Place the IPI adapter in Test mode.

5. Enable the maintenance channel's Invert Function Out Parity Bit (I4 Test mode Maintenance register test code 34B) at the CMI.

6. Initiate an IPI channel to CM data transfer in 16-bit Interlocked mode using the DMA Read To IPI ($0C00_{16}$) function.

7. Load the T register with the CM address and byte count of eight (one CM word).

8. Check, after the data transfer completes, the DMA Error Status register, CIO Channel Status register, NIO Fault Status 1 register, and CIO Fault Status 2 register for errors.

9. Disable the Invert Function Out Parity Bit at the CMI.

```
( optional error data )  MR TEST MODE =mm
                         TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 4

Tests the response on address out parity error.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select and initialize the Random Vector Generator with a PP unique data pattern.
4. Place the IPI adapter in Test mode.
5. Enable the maintenance channel's Force Address Out Parity Bit Low (I4 Test mode Maintenance register test code 35B) at the CMI.
6. Initiate an IPI channel to CM data transfer in 16-bit Interlocked mode using the DMA Read To IPI ($0C00_{16}$) function.
7. Load the T register with the CM address and byte count of eight (one CM word).
8. Check, after the data transfer completes, the DMA Error Status register, CIO Channel Status register, NIO Fault Status 1 register, and CIO Fault Status 2 register for errors.
9. Disable the Force Address Out Parity Bit Low at the CMI.

This subsection is repeated for an address bit in each parity byte.

```
( optional error data )  MR TEST MODE =mm
                         TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 5

Tests the response on tag-in/out errors.

Method of Execution:

1. Initialize the CM buffer with a PP unique data pattern.
2. Execute the IPI Master Reset sequence.
3. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
4. Select and initialize the Random Vector Generator with a data pattern.
5. Place the IPI adapter in Test mode.
6. Enable the maintenance channel's Force Tag-In/Out Parity Error (I4 Test mode Maintenance register test code 32B) at the CMI.
7. Initiate an IPI channel to CM or CM to IPI channel data transfer in 16-bit Interlocked mode using the DMA Read To IPI ($0C00_{16}$) function or the DMA Write To IPI ($0D00_{16}$) function.

8. Load the T register with the CM address and byte count of eight (one CM word).
9. Check, after the data transfer completes, the DMA Error Status register, CIO Channel Status register, NIO Fault Status 1 register, and CIO Fault Status 2 register for errors.
10. Disable the Force Tag-In/Out Parity Error at the CMI.

This subsection is repeated for transfers in both directions.

```
( optional error data )  MR TEST MODE =mm
                          TEST SEED ssss
                          XFER BLK bbbb tttt tttt
```

Subsection 6

Tests the response on mark parity bit errors.

Method of Execution:

1. Initialize the CM buffer with the PP unique data pattern.
2. Execute the IPI Master Reset sequence.
3. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
4. Select and initialize the Random Vector Generator with a data pattern.
5. Place the IPI adapter is placed in Test mode.
6. Enable the maintenance channel's Force Mark Parity Bit To Zero (I4 Test mode Maintenance register test code 31B) at the CMI.
7. Initiate an IPI channel to CM or CM to IPI channel data transfer in 16-bit Interlocked mode using the DMA Read To IPI ($0C00_{16}$) function or the DMA Write To IPI ($0D00_{16}$) function.
8. Load the T register with the CM address and byte count of eight (one CM word).
9. Check, after the data transfer completes, the DMA Error Status register, CIO Channel Status register, NIO Fault Status 1 register, and CIO Fault Status 2 register for errors.
10. Disable the Force Mark Parity Bit To Zero at the CMI.

This subsection is repeated for transfers in both directions.

```
( optional error data )  MR TEST MODE =mm
                          TEST SEED ssss
                          XFER BLK bbbb tttt tttt
```

Section 24 - DMA Transfer Test

This section is the DMA transfer test.

Subsection 0

Tests incomplete word DMA read transfers in interlocked mode.

Method of Execution:

1. Initialize the CM address that will be read.
2. Execute the IPI Master Reset sequence.
3. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
4. Select and initialize the Random Vector Generator with a PP unique data pattern.
5. Place the IPI adapter in Test mode.
6. Initiate a CM to IPI channel data transfer in 16-bit Interlocked mode using the DMA Write To IPI ($0D00_{16}$) function.
7. Load the T register with the CM address and byte count of 10 (one plus CM words).
8. Check, after the data transfer completes, the DMA Error register status and NIO Fault Status 1 register for errors.
9. Read and check the vector in the Test mode operand generator for errors.

```
( optional error data )  TEST SEED ssss
                          XFER BLK bbbb tttt tttt
```

Subsection 1

Tests incomplete word DMA write transfers in interlocked mode.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select and initialize the Random Vector Generator with a PP unique data pattern.
4. Place the IPI adapter in Test mode.
5. Initiate an IPI channel to CM data transfer in 16-bit Interlocked mode using the DMA Read From IPI ($0C00_{16}$) function.
6. Load the T register with the CM address and byte count of 10 (one plus CM words).
7. Check, after the data transfer completes, the DMA Error register status and NIO Fault Status 1 register for errors.
8. Check data in CM for errors.

```
( optional error data )  TEST SEED ssss
                         XFER BLK bbbb tttt tttt
```

Subsection 2

Tests DMA transfers in interlocked mode.

Method of Execution:

1. Execute the IPI Master Reset sequence.
2. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
3. Select and initialize the Random Vector Generator with a random data pattern.
4. Place the IPI adapter in Test mode.
5. Initiate an IPI channel to CM data transfer in 16-bit Interlocked mode using the DMA Read From IPI ($0C00_{16}$) function.
6. Load the T register with the CM address and byte count for the maximum number of CM words available.
7. Check, after the data transfer completes, the DMA Error register status and NIO Fault Status 1 register for errors.
8. Execute the IPI Master Reset sequence.
9. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
10. Select and initiate the Random Vector Generator with a random data pattern.
11. Place the IPI adapter in Test mode.
12. Initiate a CM to IPI channel data transfer in 16-bit Interlocked mode using the DMA Write To IPI ($0D00_{16}$) function.
13. Load the T register with the CM address and byte count for the maximum number of CM words available.
14. Check, after the data transfer completes, the DMA Error register status and NIO Fault Status 1 register for errors.
15. Read and check the vector in the Test mode operand generator for errors.

This subsection consists of 10 conditions each being a different random data pattern.

```
( optional error data )  TEST SEED ssss
                          XFER BLK bbbb tttt tttt
```

Subsection 3

Tests DMA transfers in streaming mode.

Method of Execution:

Same as Subsection 2, but transfers are in streaming mode.

```
( optional error data )  TEST SEED ssss
                          XFER BLK bbbb tttt tttt
```

Section 25 – Stacked Transfer Register Test

This section is the stacked transfer test.

Subsection 0

A minimum of six transfers will be sequentially sent to the IPI adapter.

Method of Execution:

1. Generate the byte counts randomly.
2. Compute the starting real memory addresses from the byte counts.
3. Execute the IPI Master Reset sequence.
4. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
5. Select and initialize the Random Vector Generator with a random data pattern.
6. Place the IPI adapter in Test mode.
7. Initiate DMA write transfers until the byte count/address pairs have expired.
8. Execute the IPI Master Reset sequence.
9. Set sync period to 20-MHz internal clock to select the 10-megabyte transfer rate.
10. Select and initialize the Random Vector Generator with a random data pattern.
11. Place the IPI adapter in Test mode.
12. Initiate DMA read transfers with the same byte count/address pairs to check the data.
13. Check the NIO Fault Status 1 and DMA Error Status registers for errors.
14. Read and check the vector in the operand generator for errors.

This sequence is repeated for all combinations of interlocked and streaming transfers.

```
( optional error data )  TEST SEED ssss
                          XFER NO.  nn
                         NXT cccc bbbb tttt tttt
                         CUR cccc bbbb tttt tttt
```

SAT4 Small Computer System Interface Adapter Channel 8

SAT4 Small Computer System Interface Adapter Channel 8

Section Descriptions

All sections of SAT4 follow the same general scheme, except for sections 00 through 02. They consist of two parts, an initiator and a target (sections 00 through 02 have only an initiator part), which are run in separate peripheral processors (PPs). Both parts of the section code are coded at a higher level by use of macros in order to save memory space (this program must fit within 8K).

Initially, each part begins in the small computer system interface (SCSI) bus disconnected state due to a local SCSI adapter Master Clear (8000_{16}) function. Then, an SBIC Reset (00_{16}) command is issued to set the Own ID for the adapter (the adapter's switches are read for use as the Own ID), the Destination ID (register 15_{16}) is set according to PARAM10 (see Parameters), and selection and/or reselection are enabled in the Source ID (16_{16}) register. At this point, the SBIC chip is interrogated to check for an "SBIC has been Selected" ($82_{16}/83_{16}$) interrupt. If the SBIC has been selected, the SBIC is in the target state and the program enters the target part. Otherwise, the program attempts to select the remote SBIC with the Timeout Period (02_{16}) register set to zero, to disable the "Select Timeout" (82_{16}) interrupt and thus enter the initiator state.

When each of the partner programs have determined the state of the SBIC (that is, initiator or target state), each program executes its role for the test sequence which is specific to the section of SAT4 being executed. In general, the initiator issues a SCSI command in a Message Out phase, which causes the target to execute a unique SCSI Bus control sequence. Finally, each sequence ends with a disconnect of the SCSI Bus, placing both SBICs in the disconnected state.

At this point, the former initiator program checks if its SBIC has been selected, while the former target attempts to select its remote SBIC. Once selection has occurred, the program continues as before, except that the initiator and target roles have been swapped between the partner PPs. When both PPs have had the opportunity to execute in both states, the subsection concludes.

Section 00 – SCSI Adapter Access Test

This section tests the basic features of the SCSI adapter that will be used in succeeding test sections.

Subsection 00

Test response to Master Clear Adapter function.

Specific hardware features tested by initiator include:

- Master Clear Adapter (8000_{16}) function

Method of Execution:

1. Sends a Master Clear Adapter (8000_{16}) function to the SCSI adapter and expects an Inactivate in response.

Subsection 01

Test response on Read Adapter Status register function.

Specific hardware features tested by initiator include:

- Read Adapter Status register (8003_{16}) function
- Differential Error (0100_{16}) status
- Write Path Parity Error (0200_{16}) status
- Zero Fill (0400_{16}) status
- Read Path Parity Error (0800_{16}) status

Method of Execution:

1. After issuing a Master Clear Adapter (8000_{16}) function, this subsection sends a Read Adapter Status (8000_{16}) function to the SCSI adapter and expects an Inactivate in response.
2. Then, it activates the channel, expects a Full, and reads one word of status before deactivating the channel.
3. Checks for bits in the adapter status that are stuck-at-one. The status of the switch 0 through 2 bits in adapter status are compared to the value in PARAM11 (see PARAMETERS).

Section 01 – SBIC Chip Register Function and Data Test

This section tests the ability of the SCSI adapter to clear, write, and read the SBIC Chip's registers.

Subsection 00

Clear and read SBIC Chip registers test.

Specific hardware features tested by initiator include:

- All SBIC Chip Register Read functions (see appendix B)
- All SBIC Chip Register functions for Stuck-at-one
- Interrupt (80_{16}) status
- Reset by MR- (00_{16}) status
- All registers are reset by (8000_{16}) function (FEATURE9)
- Own ID register (00_{16}) is reset by (8000_{16}) function (FEATURE8A)

Method of Execution:

1. Clear the local SCSI adapter (8000_{16}) and read and compare each SBIC register with zero.
2. Check the Auxiliary Status ($1F_{16}$) register for the Interrupt (80_{16}) bit cleared after the SCSI Status (17_{16}) register has been read and checked for Reset (00_{16}) status.

Subsection 01

Write and read SBIC Chip registers test.

Specific hardware features tested by initiator include:

- All SBIC Chip Register Write functions
- All SBIC Chip Register functions for Stuck-at-zero.

Method of Execution:

1. Clear the local SCSI adapter (8000_{16}).
2. Write each SBIC register with a pattern; read and compare with the pattern.
3. Repeat sequence for all patterns and all registers.

Section 02 - SBIC Chip Reset (00_{16}) Command Test

This section tests the ability of the SCSI adapter to execute a Reset (00_{16}) command in the SBIC Chip register.

Subsection 00

Test Reset (00_{16}) command operation.

Specific hardware features tested by initiator include:

- SBIC Chip Reset (00_{16}) command
- Command (18_{16}) register
- Interrupt (80_{16}) status
- Reset by Command (00_{16}) status
- Reset command (00_{16}) produces interrupt on completion (FEATURE3)

Method of Execution:

1. Clear the local SCSI adapter (8000_{16}).
2. Issue the Reset (00_{16}) command to the Command (18_{16}) register.
3. After Interrupt (80_{16}) sets in the Auxiliary Status ($1F_{16}$) register, the SCSI Status (17_{16}) register is checked for Reset by Command (00_{16}) status.

Subsection 01

Test that Reset (00_{16}) command clears all registers.

Specific hardware features tested by initiator include:

- Own ID register (00_{16}) is not reset by the reset (00_{16}) command (FEATURE8C)
- All registers are reset by Reset command (FEATURE11)

Method of Execution:

1. After the local SCSI adapter is cleared (8000_{16}), all SBIC registers are written with a unique pattern and the register under test and the Own ID (00_{16}) register are written with a (FF_{16}) pattern.
2. Issue the Reset (00_{16}) command to the Command (18_{16}) register.
3. Check, upon completion of the Reset command, the Own ID (00_{16}) register and the register under test for zero.
4. Test the Own ID (00_{16}) register for the (FF_{16}) pattern.

Section 03 – Select (07_{16}) Command and Message Out Phase Test

This section tests the ability of the SBIC Chip to operate the SCSI Channel through the Select and Message Out phases.

Subsection 00

Test Select Without Attention and Transfer command.

Specific hardware features tested by initiator include:

- SBIC Chip Attention (02_{16}) command
- SBIC Chip Select Without Attention (07_{16}) command
- SBIC Chip Single Byte Transfer ($A0_{16}$) command (output)
- A Select Command Completed Successfully (11_{16}) status
- Request Message Out Phase ($8E_{16}$) status
- A Disconnect Has Occurred (85_{16}) status
- Disconnect (04_{16}) message
- Data Buffer Ready (01_{16}) Auxiliary status (output)

Specific hardware features tested by target include:

- Receive Message Out (12_{16}) command
- Disconnect (04_{16}) command
- Enable Selection (40_{16}) in Source ID (16_{16}) register
- SBIC Has Been Selected (ATN is not asserted) (82_{16}) status
- Receive Command Completed Successfully (13_{16}) status
- ATN Signal Has Been Asserted (84_{16}) status
- Disconnect (04_{16}) message
- Data Buffer Ready (01_{16}) Auxiliary status

8

Method of Execution:

1. Clear local SCSI Adapter (8000_{16}) function.
2. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
3. Check for "Reset by Command,Power-Up,MR-" (00_{16}) in SCSI Status (17_{16}) register.
4. Set Own ID (00_{16}) register to switch settings.
5. Issue Reset (00_{16}) command.
6. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
7. Check for "Reset by Command,Power-Up,MR-" (00_{16}) in SCSI Status (17_{16}) register.
8. Set Destination ID (15_{16}) register to parameters.
9. Enable selection in Source ID (16_{16}) register.
10. Check for Target mode [that is, "SBIC Selected" (82_{16} or 83_{16}) in SCSI Status (17_{16}) register].
11. Issue Select Without Attention (07_{16}) command.
12. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
13. Check for "Select Completed Successfully" (11_{16}) in SCSI Status (17_{16}) register.
14. Issue Assert Attention (ATN) (02_{16}) command.
15. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
16. Check for "Request Message Out Phase" ($8E_{16}$) in SCSI Status (17_{16}) register.
17. Issue Single Byte Info Transfer ($A0_{16}$) command.
18. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
19. Issue Disconnect (04_{16}) message.
20. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
21. Check for "A Disconnect Has Occurred" (85_{16}) in SCSI Status (17_{16}) register.
22. Wait for timeout or interrupt.
23. Check for Target mode [that is, "SBIC Selected" (82_{16} or 83_{16}) in SCSI Status (17_{16}) register].
24. Disable selection in Source ID (16_{16}) register.
25. Repeat condition.
26. End of task.
27. End of section.

Section 04 – Select (06_{16}) Command and Message Out Phase Test

This section tests the ability of the SBIC Chip to operate the SCSI Channel through the Select and Message Out phases by using a combinational command, Select with Attention.

Subsection 00

Test Select with Attention.

Specific hardware features tested by initiator include:

- SBIC Chip Select with Attention (06_{16}) command

Specific hardware features tested by target include:

- SBIC has been Selected (ATN is asserted) (83_{16}) status

This section is identical to section 03, except that the Select with Attention (06_{16}) command is used and the target will see the Selected with Attention (83_{16}) status.

Section 05 – Remote Reset and Message Out/In Phase Test

This section tests the ability of the SBIC Chip to operate the SCSI Channel through the Message Out/Message In phases and to be reset by a Remote Reset (8004_{16}) function.

Subsection 00 – Test Select Without Attention

Specific hardware features tested by initiator include:

- SBIC Chip Single Byte Transfer ($A0_{16}$) command (input)
- Negate ACK (03_{16}) command
- Transfer Completed Successfully – Request Message In Phase ($1F_{16}$) status
- Transfer Paused (Nonmessage In Phase) with ACK Asserted (20_{16}) status
- Reset by MR- (00_{16}) status
- Bus Device Reset ($0C_{16}$) message
- Command Complete (00_{16}) message
- Data Buffer Ready (01_{16}) Auxiliary status (input)

Specific hardware features tested by target include:

- Reset Local and Remote SBIC Chips (8004_{16}) function
- SBIC Chip Send Message In (16_{16}) command
- Reset by MR- (00_{16}) status
- Bus Device Reset ($0C_{16}$) message
- Command Complete (00_{16}) message

Method of Execution:

1. Clear local SCSI adapter (8000_{16}) function.
2. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
3. Check for "Reset by Command,Power-Up,MR-" (00_{16}) in SCSI Status (17_{16}) register.
4. Set Own ID (00_{16}) register to switch settings.
5. Issue Reset (00_{16}) command.
6. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
7. Check for "Reset by Command,Power-Up,MR-" (00_{16}) in SCSI Status (17_{16}) register.
8. Set Destination ID (15_{16}) register to parameters.
9. Enable selection in Source ID (16_{16}) register.
10. Check for target mode [that is, "SBIC Selected" (82_{16} or 83_{16}) in SCSI Status (17_{16}) register].
11. Issue Select Without Attention (07_{16}) command.
12. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
13. Check for "Select Completed Successfully" (11_{16}) in SCSI Status (17_{16}) register.
14. Issue Assert Attention (ATN) (02_{16}) command.
15. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
16. Check for "Request Message Out Phase" ($8E_{16}$) in SCSI Status (17_{16}) register.
17. Issue Single Byte Info Transfer ($A0_{16}$) command.
18. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
19. Issue Bus Device Reset ($0C_{16}$) message.
20. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
21. Check for "Transfer Completed Successfully – Request Message In Phase" ($1F_{16}$) in SCSI Status (17_{16}) register.
22. Issue Single Byte Info Transfer ($A0_{16}$) command.
23. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
24. Check for "Command Complete" (00_{16}) in Data (19_{16}) register.
25. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
26. Check for "Transfer Command Paused with Acknowledge (ACK) Asserted" (20_{16}) in SCSI Status (17_{16}) register.
27. Issue Negate Acknowledge (ACK) (03_{16}) command.
28. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.

29. Check for "Reset by Command,Power-Up,MR-" (00_{16}) in SCSI Status (17_{16}) register.
30. Set Own ID (00_{16}) register to switch settings.
31. Issue Reset (00_{16}) command.
32. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
33. Check for "Reset by Command,Power-Up,MR-" (00_{16}) in SCSI Status (17_{16}) register.
34. Set Destination ID (15_{16}) register to parameters.
35. Enable selection in Source ID (16_{16}) register.
36. Wait for timeout or interrupt.
37. Check for target mode [that is, "SBIC Selected" (82_{16} or 83_{16}) in SCSI Status (17_{16}) register].
38. Disable selection in Source ID (16_{16}) register.
39. Repeat condition.
40. End of task.
41. End of section.

Section 06 – Remote Reset and Message Out/In Phase Test

This section tests the ability of the SBIC Chip to operate the SCSI Channel through the Message Out/Message In phases.

Subsection 00 – Test Select With Attention

Specific hardware features tested by initiator include:

- None unique to this section.

Specific hardware features tested by target include:

- None unique to this section.

This section is identical to section 05, except that the Select with Attention (06_{16}) command is used and the target will see the Select with Attention (83_{16}) status.

Section 07 – Single Step Commands and DBR Mode Test

This section tests the ability of the SBIC Chip to operate the SCSI Channel through all of its phases by utilizing single step commands in the target and initiator.

Specific hardware features tested by initiator include:

- Transfer Info (20_{16}) command
- SCSI Send ($0A_{16}$) command
- Transfer Completed Successfully – Request Data Out Phase (18_{16}) status
- Transfer Completed Successfully – Request Command Phase ($1A_{16}$) status
- Transfer Completed Successfully – Request Status Phase ($1B_{16}$) status
- Transfer Completed Successfully – Request Message In Phase ($1F_{16}$) status
- Timeout Period (02_{16}) register
- Good (00_{16}) Target Status in the Data (19_{16}) register
- Transfer Count (12_{16} through 14_{16}) registers
- Identify (80_{16}) message

Specific hardware features tested by target include:

- Receive Command (10_{16}) command
- Receive Data Out (11_{16}) command
- SCSI Send ($0A_{16}$) command
- Send Status In (14_{16}) command
- Transfer Count (12_{16} through 14_{16}) registers
- Identify (80_{16}) message

Method of Execution:

1. Clear local SCSI Adapter (8000_{16}) function.
2. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
3. Check status for 00_{16}) and clear interrupt.
4. Set Own ID (00_{16}) register to switch settings.
5. Issue Reset (00_{16}) command.
6. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
7. Check for "Reset by Command,Power-Up,MR-" (00_{16}) in SCSI Status (17_{16}) register.
8. Set Sync Transfer (11_{16}) register to parameters.
9. Beginning of incremental word count loop.
10. Set Timeout (02_{16}) register to infinite wait.
11. Set Destination ID (15_{16}) register to parameters.
12. Enable selection and reselection in Source ID (16_{16}) register.
13. Check for target mode [that is, "SBIC Selected" (82_{16} or 83_{16}) in SCSI Status (17_{16}) register].
14. Issue Select With Attention (06_{16}) command.
15. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
16. Check for "Select Completed Successfully" (11_{16}) in SCSI Status (17_{16}) register.
17. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
18. Check for "Request Message Out Phase" ($8E_{16}$) in SCSI Status (17_{16}) register.
19. Issue Single Byte Info Transfer ($A0_{16}$) command.
20. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
21. Issue Identify (80_{16}) message.
22. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
23. Check for "Transfer Command Completed Successfully – Request Command Phase" ($1A_{16}$) in SCSI Status (17_{16}) register.
24. Set Transfer Count (12_{16}) register to Transfer Count.
25. Set Transfer Count (13_{16}) register to Transfer Count.
26. Set Transfer Count (14_{16}) register to Transfer Count.
27. Issue Transfer Info (20_{16}) command.
28. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
29. Send SCSI Write ($0A_{16}$) command to CDB1.

30. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
31. Send Command Descriptor Block (CDB2) data.
32. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
33. Send Command Descriptor Block (CDB3) data.
34. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
35. Send Command Descriptor Block (CDB4) data.
36. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
37. Send Command Descriptor Block (CDB5) data.
38. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
39. Send Command Descriptor Block (CDB6) data.
40. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
41. Check for "Transfer Command Completed Successfully - Request Data Out Phase" (18_{16}) in SCSI Status (17_{16}) register.
42. Set Transfer Count (12_{16}) register to Transfer Count.
43. Set Transfer Count (13_{16}) register to Transfer Count.
44. Set Transfer Count (14_{16}) register to Transfer Count.
45. Issue Transfer Info (20_{16}) command.
46. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
47. Write data phase data.
48. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
49. Check for "Transfer Command Completed Successfully - Request Status Phase" ($1B_{16}$) in SCSI Status (17_{16}) register.
50. Issue Single Byte Transfer Info ($A0_{16}$) command.
51. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
52. Read SCSI status byte.
53. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
54. Check for "Transfer Command Completed Successfully - Request Message In Phase" ($1F_{16}$) in SCSI Status (17_{16}) register.
55. Issue Single Byte Transfer Info ($A0_{16}$) command.
56. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
57. Send data phase data.
58. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.

59. Check for "Transfer command paused with acknowledge (ACK) asserted" (20_{16}) in SCSI Status (17_{16}) register.

60. Issue Negate Acknowledge (ACK) (03_{16}) command.

61. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.

62. Check for "A Disconnect Has Occurred" (85_{16}) in SCSI Status (17_{16}) register.

63. Wait for timeout or interrupt.

64. Check for Target mode [that is, "SBIC Selected" (82_{16} or 83_{16}) in SCSI Status (17_{16}) register].

65. Disable selection in Source ID (16_{16}) register.

66. Repeat condition.

67. Loop until all word counts are tried.

68. End of task.

69. End of section.

Section 08 – Output Data Phase and DBR Mode Test

This section tests the ability of the SBIC Chip to operate the SCSI Channel in the output data phase.

Specific hardware features tested by initiator include:

- SBIC Chip Select with Attention and Transfer (08_{16}) command (Output)
- Transfer Completed Successfully – Request Message Out Phase ($1E_{16}$) status
- Good (00_{16}) Target Status in the Target LUN ($0F_{16}$) register
- WD-Bus Select (40_{16}) in the Control (01_{16}) register
- Command Complete Message Received (60_{16}) status in the Command Phase (10_{16}) register
- A Select and Transfer Command Completed Successfully (16_{16}) status
- Extended Message (01_{16}) message
- Disconnect and Wait For Select (80_{16}) message

Specific hardware features tested by target include:

- Wait for Select and Receive ($0C_{16}$) command
- Extended Message (01_{16}) message
- WD-Bus Select (40_{16}) in the Control (01_{16}) register
- Disconnect and Wait For Select (80_{16}) message

Method of Execution:

1. Clear local SCSI Adapter (8000_{16}) function.
2. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
3. Check for "Reset by Command,Power-Up,MR-" (00_{16}) in SCSI Status (17_{16}) register.
4. Set Own ID (00_{16}) register to switch settings.
5. Issue Reset (00_{16}) command.
6. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
7. Check for "Reset by Command,Power-Up,MR-" (00_{16}) in SCSI Status (17_{16}) register.
8. Set Sync Transfer (11_{16}) register to parameters.
9. Beginning of incremental word count loop.
10. Set Timeout (02_{16}) register to infinite wait.
11. Set Destination ID (15_{16}) register to parameters.
12. Enable selection in Source ID (16_{16}) register.
13. Check for Target mode [that is, "SBIC Selected" (82_{16} or 83_{16}) in SCSI Status (17_{16}) register].
14. Issue Select With Attention (06_{16}) command.
15. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
16. Check for "Select Completed Successfully" (11_{16}) in SCSI Status (17_{16}) register.
17. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
18. Check for "Request Message Out Phase" ($8E_{16}$) in SCSI Status (17_{16}) register.
19. Issue Single Byte Info Transfer ($A0_{16}$) command.
20. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
21. Issue Extended Message (01_{16}) message.
22. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
23. Check for "Transfer Command Completed Successfully – Request Message Out Phase" ($1E_{16}$) in SCSI Status (17_{16}) register.
24. Issue Single Byte Info Transfer ($A0_{16}$) command.
25. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
26. Issue Extended Message length equal to one.
27. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
28. Check for "Transfer Command Completed Successfully – Request Message Out Phase" ($1E_{16}$) in SCSI Status (17_{16}) register.

29. Issue Single Byte Info Transfer ($A0_{16}$) command.
30. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
31. Issue Disconnect and Wait For Select (80_{16}) message.
32. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
33. Check for "A Disconnect Has Occurred" (85_{16}) in SCSI Status (17_{16}) register.
34. Set Control (01_{16}) register for block transfer.
35. Set Timeout (02_{16}) register to maximum (2 s).
36. Set Command Descriptor Block (CDB1) (03_{16}) register to Send ($0A_{16}$) command.
37. Set Command Descriptor Block (CDB3-5) ($05\text{-}07_{16}$) registers and Transfer Count ($12\text{-}14_{16}$) registers
38. Issue Select With Attention And Transfer (08_{16}) command.
39. Output data block of "WC" words.
40. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
41. Check for "Select And Transfer Complete" (16_{16}) in SCSI Status (17_{16}) register.
42. Issue Negate Acknowledge (ACK) (03_{16}) command.
43. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
44. Check for "A Disconnect Has Occurred" (85_{16}) in SCSI Status (17_{16}) register.
45. Check for "Command Complete Message Received" (60_{16}) in Command Phase (10_{16}) register.
46. Check for Target Status equal to "Good" (00_{16}) in Target LUN/Status ($0F_{16}$) register.
47. Check for Transfer Count ($12_{16}\text{-}14_{16}$) registers equal to zero.
48. Wait for timeout or interrupt.
49. Check for Target mode [that is, "SBIC Selected" (82_{16} or 83_{16}) in SCSI Status (17_{16}) register].
50. Disable selection in Source ID (16_{16}) register.
51. Repeat condition.
52. Loop until all word counts are tried.
53. End of task.
54. End of section.

Section 09 – Input Data Phase Test

This section tests the ability of the SBIC Chip to operate the SCSI Channel in the input data phase.

Specific hardware features tested by initiator include:

- Read SCSI Data (8007_{16}) function
- SCSI Receive (08_{16}) command

Specific hardware features tested by target include:

- Send Data In (15_{16}) command
- SCSI Receive (08_{16}) command

Method of Execution:

1. Clear local SCSI Adapter (8000_{16}) function.
2. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
3. Check status for 00_{16} and clear interrupt.
4. Set OWN ID (00_{16}) register to switch settings.
5. Issue Reset (00_{16}) command.
6. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
7. Check for "Reset by Command,Power-Up,MR-" (00_{16}) in SCSI Status (17_{16}) register.
8. Set Sync Transfer (11_{16}) register to parameters.
9. Beginning of incremental word count loop.
10. Set Timeout (02_{16}) register to infinite wait.
11. Set Destination ID (15_{16}) register to parameters.
12. Enable selection in Source ID (16_{16}) register.
13. Check for Target mode [that is, "SBIC Selected" (82_{16} or 83_{16}) in SCSI Status (17_{16}) register].
14. Issue Select With Attention (06_{16}) command.
15. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
16. Check for "Select Completed Successfully" (11_{16}) in SCSI Status (17_{16}) register.
17. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
18. Check for "Request Message Out Phase" ($8E_{16}$) in SCSI Status (17_{16}) register.
19. Issue Single Byte Info Transfer ($A0_{16}$) command.
20. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.

21. Issue Extended Message (01_{16}) message.
22. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
23. Check for "Transfer Command Completed Successfully – Request Message Out Phase" ($1E_{16}$) in SCSI Status (17_{16}) register.
24. Issue Single Byte Info Transfer ($A0_{16}$) command.
25. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
26. Issue Extended Message length equal to one.
27. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
28. Check for "Transfer Command Completed Successfully – Request Message Out Phase" ($1E_{16}$) in SCSI Status (17_{16}) register.
29. Issue Single Byte Info Transfer ($A0_{16}$) command.
30. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
31. Issue Disconnect and Wait For Select (80_{16}) message.
32. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
33. Check for "A Disconnect Has Occurred" (85_{16}) in SCSI Status (17_{16}) register.
34. Set Control (01_{16}) register for block transfer.
35. Set Timeout (02_{16}) register to maximum (2 s).
36. Set Command Descriptor Block (CDB1) (03_{16}) register to Receive (08_{16}) command.
37. Set Command Descriptor Block (CDB3-5) ($05\text{-}07_{16}$) registers and Transfer Count ($12\text{-}14_{16}$) registers.
38. Prime SCSI Adapter block data input.
39. Issue Select With Attention and Transfer (08_{16}) command.
40. Input data block of "WC" words.
41. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
42. Check for "Select And Transfer Complete" (16_{16}) in SCSI Status (17_{16}) register.
43. Issue Negate Acknowledge (ACK) (03_{16}) command.
44. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
45. Check for "A Disconnect Has Occurred" (85_{16}) in SCSI Status (17_{16}) register.
46. Check for "Command Complete Message Received" (60_{16}) in Command Phase (10_{16}) register.
47. Check for Target Status equal to "Good" (00_{16}) in Target LUN/Status ($0F_{16}$) register.
48. Check for Transfer Count ($12_{16}\text{-}14_{16}$) registers equal to zero.

49. Wait for timeout or interrupt.

50. Check for Target mode [that is, "SBIC Selected" (82_{16} or 83_{16}) in SCSI Status (17_{16}) register].

51. Disable selection in Source ID (16_{16}) register.

52. Repeat condition.

53. Loop until all word counts are tried.

54. End of task.

55. End of section.

Section 10 – SBIC Chip Resume (09_{16}) Test

This section tests the ability of the SBIC Chip to operate the SCSI Channel during a resume sequence.

Specific hardware features tested by initiator include:

- Resume (09_{16}) command
- SCSI Seek ($0B_{16}$) command
- Target Disconnected (43_{16}) status in the Command Phase (10_{16}) register
- SBIC Has Been Reconnected (80_{16}) status
- Request Message In Phase After Connection ($8F_{16}$) status

Specific hardware features tested by target include:

- Reselect and Send Data ($0B_{16}$) command
- SCSI Seek ($0B_{16}$) command

Method of Execution:

1. Clear local SCSI Adapter (8000_{16}) function.
2. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
3. Check status for 00_{16}) and clear interrupt.
4. Set OWN ID (00_{16}) register to switch settings.
5. Issue Reset (00_{16}) command.
6. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
7. Check for "Reset by Command,Power-Up,MR-" (00_{16}) in SCSI Status (17_{16}) register.
8. Set Sync Transfer (11_{16}) register to parameters.
9. Beginning of incremental word count loop.
10. Set Timeout (02_{16}) register to infinite wait.
11. Set Destination ID (15_{16}) register to parameters.
12. Enable selection and reselection in Source ID (16_{16}) register.
13. Check for Target mode [that is, "SBIC Selected" (82_{16} or 83_{16}) in SCSI Status (17_{16}) register].
14. Issue Select With Attention (06_{16}) command.
15. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
16. Check for "Select Completed Successfully" (11_{16}) in SCSI Status (17_{16}) register.
17. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.

18. Check for "Request Message Out Phase" ($8E_{16}$) in SCSI Status (17_{16}) register.
19. Issue Single Byte Info Transfer ($A0_{16}$) command.
20. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
21. Issue Extended Message (01_{16}) message.
22. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
23. Check for "Transfer Command Completed Successfully – Request Message Out Phase" ($1E_{16}$) in SCSI Status (17_{16}) register.
24. Issue Single Byte Info Transfer ($A0_{16}$) command.
25. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
26. Issue Extended Message length equal to one.
27. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
28. Check for "Transfer Command Completed Successfully – Request Message Out Phase" ($1E_{16}$) in SCSI Status (17_{16}) register.
29. Issue Single Byte Info Transfer ($A0_{16}$) command.
30. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
31. Issue Disconnect and Wait For Select (80_{16}) message.
32. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
33. Check for "A Disconnect Has Occurred" (85_{16}) in SCSI Status (17_{16}) register.
34. Set Control (01_{16}) register for block transfer.
35. Set Timeout (02_{16}) register to maximum (2 s).
36. Set Command Descriptor Block (CDB1) (03_{16}) register to Seek ($0B_{16}$) command.
37. Set Command Descriptor Block (CDB3-5) ($05\text{-}07_{16}$) registers and Transfer Count ($12\text{-}14_{16}$) registers.
38. Issue Select With Attention and Transfer (08_{16}) command.
39. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
40. Check for "Target Disconnected" (43_{16}) in Command Phase (10_{16}) register.
41. Check for "A Disconnect Has Occurred" (85_{16}) in SCSI Status (17_{16}) register.
42. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
43. Check for "SBIC Has Been Reconnected" (80_{16}) in SCSI Status (17_{16}) register.
44. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
45. Check for "Request Message In Phase" ($8F_{16}$) in SCSI Status (17_{16}) register.
46. Set phase equal to "Original Target Reselected" (44_{16}).

47. Prime SCSI adapter block data input.
48. Issue Resume (09_{16}) command.
49. Input data block of "WC" words.
50. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
51. Check for "Select and Transfer Complete" (16_{16}) in SCSI Status (17_{16}) register.
52. Issue Negate Acknowledge (ACK) (03_{16}) command.
53. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
54. Check for "A Disconnect Has Occurred" (85_{16}) in SCSI Status (17_{16}) register.
55. Check for "Command Complete Message Received" (60_{16}) in Command Phase (10_{16}) register.
56. Check for Target status equal to "Good" (00_{16}) in Target LUN/Status ($0F_{16}$) register.
57. Check for Transfer Count (12_{16}-14_{16}) registers equal to zero.
58. Wait for timeout or interrupt.
59. Check for Target mode [that is, "SBIC Selected" (82_{16} or 83_{16}) in SCSI Status (17_{16}) register].
60. Disable selection in Source ID (16_{16}) register.
61. Repeat condition.
62. Loop until all word counts are tried.
63. End of task.
64. End of section.

Section 11 – Output/Input Data Compare Test

This section tests the ability of the SBIC Chip to operate the SCSI Channel to transmit data without error.

Specific hardware features tested by initiator include:

- Disconnect and Bidirectional Transfer (81_{16}) message

Specific hardware features tested by target include:

- Disconnect and Bidirectional Transfer (81_{16}) message

Method of Execution:

1. Generate data as per Parameter 14.
2. Clear local SCSI Adapter (8000_{16}) function.
3. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
4. Check status for 00_{16}) and clear interrupt.
5. Set Own ID (00_{16}) register to switch settings.
6. Issue Reset (00_{16}) command.
7. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
8. Check for "Reset by Command,Power-Up,MR-" (00_{16}) in SCSI Status (17_{16}) register.
9. Set Sync Transfer (11_{16}) register to parameters.
10. Beginning of incremental word count loop.
11. Set Timeout (02_{16}) register to infinite wait.
12. Set Destination ID (15_{16}) register to parameters.
13. Enable selection in Source ID (16_{16}) register.
14. Check for Target mode [that is, "SBIC Selected" (82_{16} or 83_{16}) in SCSI Status (17_{16}) register].
15. Issue Select With Attention (06_{16}) command.
16. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
17. Check for "Select Completed Successfully" (11_{16}) in SCSI Status (17_{16}) register.
18. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
19. Check for "Request Message Out Phase" ($8E_{16}$) in SCSI Status (17_{16}) register.
20. Issue Single Byte Info Transfer ($A0_{16}$) command.
21. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
22. Issue Extended Message (01_{16}) message.

23. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
24. Check for "Transfer Command Completed Successfully – Request Message Out Phase" ($1E_{16}$) in SCSI Status (17_{16}) register.
25. Issue Single Byte Info Transfer ($A0_{16}$) command.
26. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
27. Issue Extended Message length equal to one.
28. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
29. Check for "Transfer Command Completed Successfully – Request Message Out Phase" ($1E_{16}$) in SCSI Status (17_{16}) register.
30. Issue Single Byte Info Transfer ($A0_{16}$) command.
31. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
32. Issue Disconnect and Input/Output Data (81_{16}) message.
33. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
34. Check for "A Disconnect Has Occurred" (85_{16}) in SCSI Status (17_{16}) register.
35. Set Control (01_{16}) register for block transfer.
36. Set Timeout (02_{16}) register to maximum (2 s).
37. Set Command Descriptor Block (CDB1) (03_{16}) register to Send ($0A_{16}$) command.
38. Set Command Descriptor Block (CDB3-5) ($05\text{-}07_{16}$) registers and Transfer Count ($12\text{-}14_{16}$) registers.
39. Issue Select With Attention and Transfer (08_{16}) command.
40. Output data block of "WC" words.
41. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
42. Check for "Select and Transfer Complete" (16_{16}) in SCSI Status (17_{16}) register.
43. Issue Negate Acknowledge (ACK) (03_{16}) command.
44. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
45. Check for "A Disconnect Has Occurred" (85_{16}) in SCSI Status (17_{16}) register.
46. Check for "Command Complete Message Received" (60_{16}) in Command Phase (10_{16}) register.
47. Check for Target Status equal to "Good" (00_{16}) in Target LUN/Status ($0F_{16}$) register.
48. Check for Transfer Count ($12_{16}\text{-}14_{16}$) registers equal to zero.
49. Set Control (01_{16}) register for block transfer.
50. Set Timeout (02_{16}) register to maximum (2 s).

51. Set Command Descriptor Block (CDB1) (03_{16}) register to Receive (08_{16}) command.
52. Set Command Descriptor Block (CDB3-5) ($05\text{-}07_{16}$) registers and Transfer Count ($12\text{-}14_{16}$) registers.
53. Prime SCSI Adapter block data input.
54. Issue Select With Attention and Transfer (08_{16}) command.
55. Input data block of "WC" words.
56. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
57. Check for "Select And Transfer Complete" (16_{16}) in SCSI Status (17_{16}) register.
58. Issue Negate Acknowledge (ACK) (03_{16}) command.
59. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
60. Check for "A Disconnect Has Occurred" (85_{16}) in SCSI Status (17_{16}) register.
61. Check for "Command Complete Message Received" (60_{16}) in Command Phase (10_{16}) register.
62. Check for Target status equal to "Good" (00_{16}) in Target LUN/Status ($0F_{16}$) register.
63. Check for Transfer Count ($12_{16}\text{-}14_{16}$) registers equal to zero.
64. Compare "WC" words of input buffer with output buffer.
65. Wait for timeout or interrupt.
66. Check for Target mode [that is, "SBIC Selected" (82_{16} or 83_{16}) in SCSI Status (17_{16}) register].
67. Disable selection in Source ID (16_{16}) register.
68. Repeat condition.
69. Loop until all word counts are tried.
70. End of task.
71. End of section.

Section 12 – Select Phase Zero Parity Test

This section tests the ability of the SBIC Chip to generate and check parity on the SCSI bus during the selection phase.

Specific hardware features tested by initiator include:

- Select Write Parity Error (85_{16}) message
- Write Zero Parity (8005_{16}) function
- A Time-out During Select (42_{16}) status

Specific hardware features tested by target include:

- Select Write Parity Error (85_{16}) message

Method of Execution:

1. Clear local SCSI Adapter (8000_{16}) function.
2. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
3. Check for "Reset by Command,Power-UP,MR-" (00_{16}) in SCSI Status (17_{16}) register and clear interrupt.
4. OWN ID (00_{16}) register to switch settings.
5. Issue Reset (00_{16}) command.
6. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
7. Check for "Reset by Command,Power-UP,MR-" (00_{16}) in SCSI Status (17_{16}) register and clear interrupt.
8. Sync Transfer (11_{16}) register to parameters.
9. Set Timeout (02_{16}) register to infinite wait.
10. Set Destination ID (15_{16}) register to parameters.
11. Enable selection and reselection in Source ID (16_{16}) register.
12. Check for Target mode [that is, "SBIC Selected" (82_{16} or 83_{16}) in SCSI Status (17_{16}) register].
13. Issue Select With Attention (06_{16}) command.
14. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
15. Check for "Select Completed Successfully" (11_{16}) in SCSI Status (17_{16}) register.
16. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
17. Check for "Request Message Out Phase" ($8E_{16}$) in SCSI Status (17_{16}) register.
18. Issue Single Byte Info Transfer ($A0_{16}$) command.
19. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.

20. Issue Extended Message (01_{16}) message.
21. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
22. Check for "Transfer Command Completed Successfully – Request Message Out Phase" ($1E_{16}$) in SCSI Status (17_{16}) register.
23. Issue Single Byte Info Transfer ($A0_{16}$) command.
24. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
25. Issue Extended Message length equal to one.
26. Wait for interrupt in Auxiliary Status ($1F_{16}$) register. Check for "Transfer Command Completed Successfully – Request.
27. Message Out Phase" ($1E_{16}$) in SCSI Status (17_{16}) register. Issue Single Byte Info Transfer ($A0_{16}$) command.
28. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
29. Issue Select Write Parity Error (85_{16}, SLWPE) message.
30. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
31. Check for "A Disconnect Has Occurred" (85_{16}) in SCSI Status (17_{16}) register.
32. Set Timeout (02_{16}) register to maximum (2 s).
33. Set Destination ID (15_{16}) register to parameters.
34. Enable selection and reselection in Source ID (16_{16}) register.
35. Set write zero parity on SCSI channel (8005_{16}) function.
36. Issue Select With Attention (06_{16}) command.
37. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
38. Check for "A Time-out During Select" (42_{16}) in SCSI Status (17_{16}) register.
39. Clear local SCSI Adapter (8000_{16}) function.
40. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
41. Check for "Reset by Command,Power-UP,MR-" (00_{16}) in SCSI Status (17_{16}) register and clear interrupt.
42. Set OWN ID (00_{16}) register to switch settings.
43. Issue Reset (00_{16}) command.
44. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
45. Check for "Reset by Command,Power-UP,MR-" (00_{16}) in SCSI Status (17_{16}) register and clear interrupt.
46. Set Sync Transfer (11_{16}) register to parameters.
47. Set Timeout (02_{16}) register to maximum (2 s).

48. Set Destination ID (15_{16}) register to parameters.
49. Enable selection and reselection in Source ID (16_{16}) register.
50. Issue Select With Attention (06_{16}) command.
51. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
52. Check for "Select Completed Successfully" (11_{16}) in SCSI Status (17_{16}) register.
53. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
54. Check for "Request Message Out Phase" ($8E_{16}$) in SCSI Status (17_{16}) register.
55. Issue Single Byte Info Transfer ($A0_{16}$) command.
56. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
57. Issue Disconnect (04_{16}) message.
58. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
59. Check for "A Disconnect Has Occurred" (85_{16}) in SCSI Status (17_{16}) register.
60. Wait for timeout or interrupt.
61. Check for Target mode [that is, "SBIC Selected" (82_{16} or 83_{16}) in SCSI Status (17_{16}) register].
62. Disable selection in Source ID (16_{16}) register.
63. Repeat condition.
64. End of task.
65. End of section.

Section 13 – Message/Command/Data Out Phase Zero Parity Test

This section tests the ability of the SBIC Chip to generate and check parity on the SCSI bus during the output phases.

Specific hardware features tested by initiator include:

- Message Out Write Parity Error (82_{16}) message (condition 0)
- Command Descriptor Write Parity Error (83_{16}) message (condition 1)
- Data Out Write Parity Error (84_{16}) message (condition 2)
- Identify with Parity Error (81_{16}) message

Specific hardware features tested by target include:

- Message Out Write Parity Error (82_{16}) message (condition 0)
- Command Descriptor Write Parity Error (83_{16}) message (condition 1)
- Data Out Write Parity Error (84_{16}) message (condition 2)
- Identify with Parity Error (81_{16}) message
- Halt on Parity Error (01_{16}) in the Control (01_{16}) register
- 6 CDB Transfers Made (36_{16}) in the Command Phase (10_{16}) register
- Parity Error (02_{16}) status

Method of Execution:

1. Clear local SCSI Adapter (8000_{16}) function.
2. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
3. Check for "Reset by Command,Power-UP,MR-" (00_{16}) in SCSI Status (17_{16}) register and clear interrupt.
4. Set OWN ID (00_{16}) register to switch settings.
5. Issue Reset (00_{16}) command.
6. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
7. Check for "Reset by Command,Power-UP,MR-" (00_{16}) in SCSI Status (17_{16}) register and clear interrupt.
8. Set Sync Transfer (11_{16}) register to parameters.
9. Begin output phase write parity error loop.
10. Set Timeout (02_{16}) register to infinite wait.
11. Set Destination ID (15_{16}) register to parameters.
12. Enable selection and reselection in Source ID (16_{16}) register.
13. Check for Target mode [that is, "SBIC Selected" (82_{16} or 83_{16}) in SCSI Status (17_{16}) register].
14. Issue Select With Attention (06_{16}) command.
15. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
16. Check for "Select Completed Successfully" (11_{16}) in SCSI Status (17_{16}) register.
17. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
18. Check for "Request Message Out Phase" ($8E_{16}$) in SCSI Status (17_{16}) register.
19. Issue Single Byte Info Transfer ($A0_{16}$) command.
20. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
21. Issue Extended Message (01_{16}) message.
22. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
23. Check for "Transfer Command Completed Successfully – Request Message Out Phase" ($1E_{16}$) in SCSI Status (17_{16}) register.
24. Issue Single Byte Info Transfer ($A0_{16}$) command.
25. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
26. Issue Extended Message length equal to one.
27. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.

28. Check for "Transfer Command Completed Successfully – Request Message Out Phase" ($1E_{16}$) in SCSI Status (17_{16}) register.
29. Issue Single Byte Info Transfer ($A0_{16}$) command.
30. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
31. Issue output phase write parity error message from "DS1300" [Condition 0=Message Out (82_{16}, MOWPE); Condition 1=Command Descriptor (83_{16}, DBWPE); Condition 2=Data Out (84_{16}, DOWPE)].
32. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
33. Check for "A Disconnect Has Occurred" (85_{16}) in SCSI Status (17_{16}) register.
34. Set Timeout (02_{16}) register to maximum (2 s).
35. Set Destination ID (15_{16}) register to parameters.
36. Enable selection and reselection in Source ID (16_{16}) register.
37. Issue Select With Attention (06_{16}) command.
38. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
39. Check for "Select Completed Successfully" (11_{16}) in SCSI Status (17_{16}) register.
40. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
41. Check for "Request Message Out Phase" ($8E_{16}$) in SCSI Status (17_{16}) register.
42. Set write zero parity on SCSI channel (8005_{16}) function for condition 0 only.
43. Issue Single Byte Info Transfer ($A0_{16}$) command.
44. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
45. Issue Identify (80_{16} for conditions 1 and 2 or 81_{16} for condition 0) message.
46. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
47. Check for "Transfer Command Completed Successfully – Request Command Phase" ($1A_{16}$) in SCSI Status (17_{16}) register.
48. Set Transfer Count (14_{16}) register to transfer count.
49. Set write zero parity on SCSI channel (8005_{16}) function for condition 1 only.
50. Issue Transfer Info (20_{16}) command.
51. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
52. Send SCSI Write ($0A_{16}$) command to CDB1 (causes parity error in condition 1).
53. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
54. Send Command Descriptor Block (CDB2) data.
55. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.

56. Send Command Descriptor Block (CDB3) data.

57. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.

58. Send Command Descriptor Block (CDB4) data.

59. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.

60. Send Command Descriptor Block (CDB5) data.

61. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.

62. Send Command Descriptor Block (CDB6) data If not data out parity error condition, go to S13RS00.

63. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.

64. Check for "Transfer Command Completed Successfully – Request Data Out Phase" (18_{16}) in SCSI Status (17_{16}) register.

65. Set write zero parity on SCSI channel (8005_{16}) function for condition 2 only.

66. Issue Single Byte Info Transfer ($A0_{16}$) command.

67. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.

68. Write data out phase data (00_{16}) with parity error S13RS00.

69. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.

70. Check for "Transfer Command Completed Successfully – Request Message Out Phase" ($1E_{16}$) in SCSI Status (17_{16}) register.

71. Issue Single Byte Info Transfer ($A0_{16}$) command.

72. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.

73. Issue Bus Device Reset ($0C_{16}$) message.

74. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.

75. Check for "Transfer Command Completed Successfully – Request Message In Phase" ($1F_{16}$) in SCSI Status (17_{16}) register.

76. Issue Single Byte Info Transfer ($A0_{16}$) command.

77. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.

78. Check for "Command Complete" (00_{16}) message in Data (19_{16}) register.

79. Wait for interrupt and parity error in Auxiliary Status ($1F_{16}$) register.

80. Check for "Transfer Command Paused With Acknowledge (ACK) Asserted" (20_{16}) in SCSI Status (17_{16}) register.

81. Issue Negate Acknowledge (ACK) (03_{16}) command.

82. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.

83. Check for "Reset by Command,Power-UP,MR-" (00_{16}) in SCSI Status (17_{16}) register and clear interrupt.

84. Set OWN ID (00_{16}) register to switch settings.

85. Issue Reset (00_{16}) command.

86. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.

87. Check for "Reset by Command,Power-UP,MR-" (00_{16}) in SCSI Status (17_{16}) register and clear interrupt.

88. Set Destination ID (15_{16}) register to parameters.

89. Enable selection in Source ID (16_{16}) register.

90. Wait for timeout or interrupt.

91. Check for Target mode [that is, "SBIC Selected" (82_{16} or 83_{16}) in SCSI Status (17_{16}) register].

92. Disable selection in Source ID (16_{16}) register.

93. Repeat condition.

94. Loop until all output phase write parity error conditions are tested.

95. End of task.

96. End of section.

Section 14 – Data/Status/Message In Phase Zero Parity Test

This section tests the ability of the SBIC Chip to generate and check parity on the SCSI bus during the input phases.

Specific hardware features tested by initiator include:

- Data In Write Parity Error (86_{16}) message (condition 0)
- Status In Write Parity Error (87_{16}) message (condition 1)
- Message In Write Parity Error (88_{16}) message (condition 2)
- Parity Error (02_{16}) status

Specific hardware features tested by target include:

- Parity Error (86_{16}) message (condition 0)
- Status In Write Parity Error (87_{16}) message (condition 1)
- Message In Write Parity Error (88_{16}) message (condition 2)
- Write Zero Parity (8005_{16}) function
- Intermediate Good (10_{16}) Target status

Method of Execution:

1. Set transfer count to two bytes for all block transfers in this section.
2. Clear local SCSI Adapter (8000_{16}) function.
3. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
4. Check for "Reset by Command,Power-UP,MR-" (00_{16}) in SCSI Status (17_{16}) register and clear interrupt.
5. Set OWN ID (00_{16}) register to switch settings.
6. Issue Reset (00_{16}) command.
7. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
8. Check for "Reset by Command,Power-UP,MR-" (00_{16}) in SCSI Status (17_{16}) register and clear interrupt.
9. Set Sync Transfer (11_{16}) register to parameters.
10. Begin input phase write parity error loop.
11. Set Timeout (02_{16}) register to infinite wait.
12. Set Destination ID (15_{16}) register to parameters.
13. Enable selection and reselection in Source ID (16_{16}) register.
14. Check for Target mode [that is, "SBIC Selected" (82_{16} or 83_{16}) in SCSI Status (17_{16}) register].
15. Issue Select With Attention (06_{16}) command.
16. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
17. Check for "Select Completed Successfully" (11_{16}) in SCSI Status (17_{16}) register.
18. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
19. Check for "Request Message Out Phase" ($8E_{16}$) in SCSI Status (17_{16}) register.
20. Issue Single Byte Info Transfer ($A0_{16}$) command.
21. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
22. Issue Extended Message (01_{16}) message.
23. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
24. Check for "Transfer Command Completed Successfully – Request Message Out Phase" ($1E_{16}$) in SCSI Status (17_{16}) register.
25. Issue Single Byte Info Transfer ($A0_{16}$) command.
26. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
27. Issue Extended Message length equal to one.

28. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
29. Check for "Transfer Command Completed Successfully - Request Message Out Phase" ($1E_{16}$) in SCSI Status (17_{16}) register.
30. Issue Single Byte Info Transfer ($A0_{16}$) command.
31. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
32. Issue input phase write parity error message from "DS1400" [Condition 0=Data In (86_{16}, DIWPE); Condition 1=Status In (87_{16}, DBWPE); Condition 2=Message In (88_{16}, MIWPE)].
33. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
34. Check for "A Disconnect Has Occurred" (85_{16}) in SCSI Status (17_{16}) register.
35. Set Control (01_{16}) register for block transfer.
36. Set Timeout (02_{16}) register to maximum (2 s).
37. Set CDB 1 (03_{16}) register to Receive (08_{16}) command.
38. Set CDB3-5 ($05\text{-}07_{16}$) and transfer count ($12\text{-}14_{16}$).
39. Prime SCSI adapter block data input (8007_{16}) function.
40. Issue Select With Attention And Transfer (08_{16}) command.
41. Input data.
42. Wait for interrupt and parity error in Auxiliary Status ($1F_{16}$) register.
43. Check for "Select and Transfer Complete" (16_{16}) in SCSI Status (17_{16}) register.
44. Check for "Command Complete Message Received" (60_{16}) in Command Phase (10_{16}) register.
45. Wait for interrupt and parity error in Auxiliary Status ($1F_{16}$) register.
46. Check for "Request Message Out Phase" ($8E_{16}$) in SCSI Status (17_{16}) register.
47. Issue Single Byte Info Transfer ($A0_{16}$) command.
48. Wait for data buffer ready and parity error in Auxiliary Status ($1F_{16}$) register.
49. Issue Bus Device Reset ($0C_{16}$) message.
50. Wait for interrupt and parity error in Auxiliary Status ($1F_{16}$) register.
51. Check for "Transfer Command Completed Successfully - Request Message In Phase" ($1F_{16}$) in SCSI Status (17_{16}) register.
52. Issue Single Byte Info Transfer ($A0_{16}$) command.
53. Wait for data buffer ready and parity error in Auxiliary Status ($1F_{16}$) register.
54. Check for "Command Complete" (00_{16}) message in Data (19_{16}) register.
55. Wait for interrupt and parity error in Auxiliary Status ($1F_{16}$) register.

56. Check for "Transfer Command Paused With Acknowledge (ACK) Asserted" (20_{16}) in SCSI Status (17_{16}) register.

57. Issue Negate Acknowledge (ACK) (03_{16}) command.

58. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.

59. Check for "Reset by Command,Power-UP,MR-" (00_{16}) in SCSI Status (17_{16}) register and clear interrupt.

60. OWN ID (00_{16}) register to switch settings.

61. Issue Reset (00_{16}) command.

62. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.

63. Check for "Reset by Command,Power-UP,MR-" (00_{16}) in SCSI Status (17_{16}) register and clear interrupt.

64. Set Destination ID (15_{16}) register to parameters.

65. Enable selection in Source ID (16_{16}) register.

66. Wait for timeout or interrupt.

67. Check for Target mode [that is, "SBIC Selected" (82_{16} or 83_{16}) in SCSI Status (17_{16}) register].

68. Disable selection in Source ID (16_{16}) register.

69. Repeat condition.

70. Loop until all input phase write parity error conditions are tested.

71. End of task.

72. End of section.

Target Program Description

Since the same target program is used for all sections in SAT4, this section has been provided to describe the target program in only one place. The hardware features tested are not listed here because each section uses a different portion of the target program (see the pertinent section for the features tested).

The initial target routine, that is entered when "SBIC has been Selected" (82_{16} or 83_{16}) status is detected, clears the Control (01_{16}) register and issues a Receive Message Out (12_{16}) command. It then waits for Data Buffer Ready (01_{16}) Auxiliary status after which it reads the Message Out data from the Data (19_{16}) register. The Message Out data is used to determine which routine to execute. If the Message Out data code is not in the table below, the error message "UNDEFINED SCSI MESSAGE CODE" is displayed.

MSG OUT CODE	ROUTINE NAME	ROUTINE TITLE
01	TXTMSG	EXTENDED MESSAGE Task
04	TDSCON	DISCONNECT Task
10	TNOOPR	NO OPERATION Task
14	TBDRST	BUS DEVICE RESET Task
80	TIDMSG	ID MESSAGE Task

The previous tasks are described in the following paragraphs.

Disconnect (04_{16}) Task

Method of Execution:

1. Wait for a "Receive Command Completed Successfully" (13_{16}) interrupt.
2. Issue a Disconnect (04_{16}) command.
3. Return control to the section code from which it was called into the target driver.

No Operation (10_{16}) Task

Method of Execution:

1. Wait for a "Receive Command Completed Successfully" (13_{16}) interrupt.
2. Issue a Disconnect (04_{16}) command.
3. Return control to the section code from which it was called into the target driver.

Bus Device Reset (14_{16}) Task

Method of Execution:

1. Wait for a "Receive Command Completed Successfully" (13_{16}) interrupt.
2. Issue a Send Message In (16_{16}) command.
3. After Data Buffer Ready (01_{16}) status sets, it places the "Command Complete" (00_{16}) SCSI message in the Data (19(6)) register and waits for "Send Command Completed Successfully" (13_{16}) interrupt.
4. Issue a Disconnect (04_{16}) command.
5. Issue Reset SCSI Channel and SCSI Adapter (8004_{16}) function, and wait for a Reset by Command (00_{16}) interrupt.
6. Before returning control to the section code from which it was called into the target driver, it uses the Reset (00_{16}) command to set the Own ID (00_{16}) register according to the Adapter's switches and sets the Destination ID (15_{16}) register according to parameter, PARAM10.

ID Message (80_{16}) Task

Method of Execution:

1. Wait for a "Receive Command Completed Successfully" (13_{16}) interrupt.
2. Set the Transfer Count (12_{16}-14_{16}) registers for 6 bytes.
3. Issue a Receive Command (10_{16}) command.
4. Read each word of the Command Descriptor Block from the Data (19_{16}) register each time Data Buffer Ready sets (the data expected and checked is $0A_{16}$ through $0F_{16}$), and waits for a second "Receive Command Completed Successfully" (13_{16}) interrupt.
5. Set the Transfer Count (12_{16}-14_{16}) registers for one byte.
6. Issue a Receive Data Out (11_{16}) command.
7. Read a word of zeroes from the Data (19_{16}) register when Data Buffer Ready sets.
8. Wait for a "Receive Command Completed Successfully" (13_{16}) interrupt.
9. Issue a Send Status (14_{16}) command.
10. Set the status in the Data (19_{16}) register after Data Buffer Ready sets, and wait for "Send Command Completed Successfully" (13_{16}) interrupt.
11. Issue a Send Message In (16_{16}) command.
12. Set the Command Complete (00_{16}) Message in the Data (19_{16}) register after Data Buffer Ready sets, and wait for "Send Command Completed Successfully" (13_{16}) interrupt.
13. Issue a Disconnect (04_{16}) command.

Extended Message (01_{16}) Task

Method of Execution:

1. Wait for a "Receive Command Completed Successfully" (13_{16}) interrupt.
2. Issue a Receive Message Out (12_{16}) command in order to read the length of the Extended Message which must be one.
3. After receiving the second "Receive Command Completed Successfully" (13_{16}) interrupt, another Receive Message Out (12_{16}) command is issued and the Vendor Unique Extended Message code ($8X_{16}$) is read.

The Extended Message code is used to determine which routine to execute. If the Extended Message code is not in the table below, the error message "UNDEFINED EXTENDED MESSAGE CODE" is displayed.

EXT MSG CODE	ROUTINE NAME	ROUTINE TITLE
80	TDSFWS	DISCONNECT, RECONNECT, AND TRANSFER Task
81	TINOUT	DISCONNECT, INPUT, AND OUTPUT Task
82	TMOWPE	Disconnect, Wait for Select and Receive, and expect a Message Out write parity error
83	TCDWPE	Disconnect, Wait for Select and Receive, and expect a Command Descriptor Block write parity error
84	TDOWPE	Disconnect, Wait for Select and Receive, and expect a Data Out write parity error
85	TSLWPE	Disconnect, Wait for Select parity error and receive message out
86	TDIWPE	Disconnect, Wait for Select and Receive, and force data in write parity error
87	TSIWPE	Disconnect, Wait for Select and Receive, and force status in write parity error
88	TMIWPE	Disconnect, Wait for Select and Receive, and force Message In write parity error

The previous routines (tasks) are described in the following paragraphs.

Disconnect, Input, and Output (81_{16}) Task

Method of Execution:

1. Wait for a "Receive Command Completed Successfully" (13_{16}) interrupt.
2. Issue a Disconnect (04_{16}) command.
3. Issue a Wait for Select and Receive ($0C_{16}$) command.
4. Wait for a "Wait for Select and Receive Command Completed Successfully (13_{16}) interrupt.
5. Read the SCSI Command code from the CDB 01 (03_{16}) register and check it for a Send ($0A_{16}$) command.
6. Set WDB (40_{16}) in the Control (01_{16}) register and use the CDB (05_{16} through 07_{16}) registers in order to set the Transfer Count (12_{16} through 14_{16}) registers.

7. Issue a Receive Data (11_{16}) command.
8. Input the appropriate number of bytes of data.
9. Wait for "Receive Command Completed Successfully" (13_{16}) interrupt.
10. Issue a Send Status (14_{16}) command, sets the status in the Data (19_{16}) register after Data Buffer Ready sets, and waits for "Send Command Completed Successfully" (13_{16}) interrupt.
11. Issue a Send Message In (16_{16}) command to finish the Send sequence.
12. Set the Command Complete (00_{16}) Message in the Data (19_{16}) register after Data Buffer Ready sets, and wait for "Send Command Completed Successfully" (13_{16}) interrupt.
13. Issue a Disconnect (04_{16}) command.
14. Continue by issuing a Wait for Select and Receive ($0C_{16}$) command.
15. Wait for a "Wait for Select and Receive Command Completed Successfully" (13_{16}) interrupt.
16. Read the SCSI Command code from the CDB 01 (03_{16}) register and check it for a Receive (08_{16}) command.
17. Set WDB (40_{16}) in the Control (01_{16}) register and use the CDB (05_{16} through 07_{16}) registers in order to set the Transfer Count (12_{16} through 14_{16}) registers.
18. Issue a Send Data In (15_{16}) command.
19. Output the appropriate number of bytes of data, and wait for "Send Command Completed Successfully" (13_{16}) interrupt.
20. Issue a Send Status (14_{16}) command.
21. Set the status in the Data (19_{16}) register after Data Buffer Ready sets, and wait for "Send Command Completed Successfully" (13_{16}) interrupt.
22. Issue a Send Message In (16_{16}) command.
23. Set the Command Complete (00_{16}) Message in the Data (19_{16}) register after Data Buffer Ready sets, and wait for "Send Command Completed Successfully" (13_{16}) interrupt.
24. Issue a Disconnect (04_{16}) command.

Message Out Write Parity Error (82_{16}) Task

Method of Execution:

1. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
2. Check for "Receive Command Completed Successfully" (13_{16}) in SCSI Status (17_{16}) register.
3. Issue Disconnect (04_{16}) command.
4. Set Control (01_{16}) register for halt on parity error.
5. Issue Wait For Select And Receive ($0C_{16}$) command.
6. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
7. Check for "Wait For Select And Receive Command Completed Successfully" (13_{16}) in SCSI Status (17_{16}) register.
8. Check for "6 CDB Transfers Made" (36_{16}) in the Command Phase (10_{16}) register.
9. Check for "Identify Message W/Parity Error" (81_{16}) message in the Target LUN ($0F_{16}$) register.
10. Go to "TARGET" routine to receive Message Out that will ask for a Bus Device Reset.

Command Descriptor Write Parity Error (83_{16}) Task

Method of Execution:

1. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
2. Check for "Receive Command Completed Successfully" (13_{16}) in SCSI Status (17_{16}) register.
3. Issue Disconnect (04_{16}) command.
4. Set Control (01_{16}) register for halt on parity error.
5. Issue Wait For Select And Receive ($0C_{16}$) command.
6. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
7. Check for "Wait For Select And Receive Command Completed Successfully" (13_{16}) in SCSI Status (17_{16}) register.
8. Check for "6 CDB Transfers Made" (36_{16}) in the Command Phase (10_{16}) register.
9. Go to "TARGET" routine to receive Message Out that will ask for a Bus Device Reset.

Data Out Write Parity Error (84_{16}) Task

Method of Execution:

1. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
2. Check for "Receive Command Completed Successfully" (13_{16}) in SCSI Status (17_{16}) register.
3. Issue Disconnect (04_{16}) command.
4. Set Control (01_{16}) register for halt on parity error.
5. Issue Wait For Select And Receive ($0C_{16}$) command.
6. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
7. Check for "Wait For Select And Receive Command Completed Successfully" (13_{16}) in SCSI Status (17_{16}) register.
8. Check for "6 CDB Transfers Made" (36_{16}) in the Command Phase (10_{16}) register.
9. Set Transfer Count (14_{16}) register to transfer count.
10. Issue Receive Data Out (11_{16}) command.
11. Wait for data buffer ready and parity error in Auxiliary Status ($1F_{16}$) register.
12. Check for parity error in Auxiliary Status ($1F_{16}$) register.
13. Read and Check data with parity error.
14. Wait for interrupt and parity error in Auxiliary Status ($1F_{16}$) register.
15. Check for "Receive Command Completed Successfully" (13_{16}) in SCSI Status (17_{16}) register.
16. Go to "TARGET" routine to receive Message Out that will ask for a Bus Device Reset.

Select Write Parity Error (85_{16}) Task

Method of Execution:

1. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
2. Check for "Receive Command Completed Successfully" (13_{16}) in SCSI Status (17_{16}) register.
3. Issue Disconnect (04_{16}) command.
4. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
5. Check for "SBIC Selected W/ATN" (83_{16}) in SCSI Status (17_{16}) register.
6. Clear Control (01_{16}) register.
7. Issue Receive Message Out (12_{16}) command.
8. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
9. Read Message Out data.
10. Check Message Out for desired operation.

Data In Write Parity Error (86_{16}) Task

Method of Execution:

1. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
2. Check for "Receive Command Completed Successfully" (13_{16}) in SCSI Status (17_{16}) register.
3. Issue Disconnect (04_{16}) command.
4. Issue Wait For Select And Receive ($0C_{16}$) command.
5. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
6. Check for "Wait For Select And Receive Command Completed Successfully" (13_{16}) in SCSI Status (17_{16}) register.
7. Check for "6 CDB Transfers Made" (36_{16}) in the Command Phase (10_{16}) register.
8. Read SCSI command from CDB 01 (03_{16}) register.
9. Use CDB (05_{16}) through (07_{16}) in order to set the Transfer Count (12_{16}) through (14_{16}) registers.
10. Set write zero parity on SCSI channel (8005_{16}) function.
11. Issue Send Data In (15_{16}) command.
12. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
13. Send word that does not require parity.
14. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.

15. Send word that does require parity.
16. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
17. Check for "Send Completed Successfully" (13_{16}) in SCSI Status (17_{16}) register.
18. Issue Send Status (14_{16}) command.
19. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
20. Issue "Intermediate Good" (10_{16}) status.
21. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
22. Check for "Send Completed Successfully" (13_{16}) in SCSI Status (17_{16}) register.
23. Issue Send Message In (16_{16}) command.
24. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
25. Issue "Command Complete" (00_{16}) message.
26. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
27. Check for "Send Completed Successfully" (13_{16}) in SCSI Status (17_{16}) register.
28. Go to "TARGET" routine to receive Message Out that will ask for a Bus Device Reset.

Status In Write Parity Error (87_{16}) Task

Method of Execution:

1. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
2. Check for "Receive Command Completed Successfully" (13_{16}) in SCSI Status (17_{16}) register.
3. Issue Disconnect (04_{16}) command.
4. Issue Wait For Select And Receive ($0C_{16}$) command.
5. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
6. Check for "Wait For Select And Receive Command Completed Successfully" (13_{16}) in SCSI Status (17_{16}) register.
7. Check for "6 CDB Transfers Made" (36_{16}) in the Command Phase (10_{16}) register.
8. Read SCSI command from CDB 01 (03_{16}) register.
9. Use CDB (05_{16}) through (07_{16}) in order to set the Transfer Count (12_{16}) through (14_{16}) registers.
10. Issue Send Data In (15_{16}) command.
11. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
12. Send word that does not require parity.
13. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
14. Send word that does require parity.
15. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
16. Check for "Send Completed Successfully" (13_{16}) in SCSI Status (17_{16}) register.

17. Set write zero parity on SCSI channel (8005_{16}) function.
18. Issue Send Status (14_{16}) command.
19. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
20. Issue "Command Complete" (00_{16}) message.
21. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
22. Check for "Send Completed Successfully" (13_{16}) in SCSI Status (17_{16}) register.
23. Issue Send Message In (16_{16}) command.
24. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
25. Issue "Command Complete" (00_{16}) message.
26. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
27. Check for "Send Completed Successfully" (13_{16}) in SCSI Status (17_{16}) register.
28. Go to "TARGET" routine to receive Message Out that will ask for a Bus Device Reset.

Message In Write Parity Error (88_{16}) Task

Method of Execution:

1. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
2. Check for "Receive Command Completed Successfully" (13_{16}) in SCSI Status (17_{16}) register.
3. Issue Disconnect (04_{16}) command.
4. Issue Wait For Select And Receive ($0C_{16}$) command.
5. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
6. Check for "Wait For Select And Receive Command Completed Successfully" (13_{16}) in SCSI Status (17_{16}) register.
7. Check for "6 CDB Transfers Made" (36_{16}) in the Command Phase (10_{16}) register.
8. Read SCSI command from CDB 01 (03_{16}) register.
9. Use CDB (05_{16}) through (07_{16}) in order to set the Transfer Count (12_{16}) through (14_{16}) registers.
10. Issue Send Data In (15_{16}) command.
11. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
12. Send word that does not require parity.
13. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
14. Send word that does require parity.
15. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
16. Check for "Send Completed Successfully" (13_{16}) in SCSI Status (17_{16}) register.

17. Issue Send Status (14_{16}) command.
18. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
19. Issue "Command Complete" (00_{16}) message.
20. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
21. Check for "Send Completed Successfully" (13_{16}) in SCSI Status (17_{16}) register.
22. Set write zero parity on SCSI channel (8005_{16}) function.
23. Issue Send Message In (16_{16}) command.
24. Wait for Data Buffer Ready in Auxiliary Status ($1F_{16}$) register.
25. Issue "Command Complete" (00_{16}) message.
26. Wait for interrupt in Auxiliary Status ($1F_{16}$) register.
27. Check for "Send Completed Successfully" (13_{16}) in SCSI Status (17_{16}) register.
28. Go to "TARGET" routine to receive Message Out that will ask for a Bus Device Reset.

Disconnect, Reconnect, and Transfer (80_{16}) Task

Method of Execution:

1. Wait for a "Receive Command Completed Successfully" (13_{16}) interrupt.
2. Issue a Disconnect (04_{16}) command.
3. Issue a Wait for Select and Receive ($0C_{16}$) command.
4. Wait for a "Wait for Select and Receive Command Completed Successfully" (13_{16}) interrupt.
5. Read the SCSI Command code from the CDB 01 (03_{16}) register.

The SCSI Command code is used to determine which routine to execute. If the SCSI Command code is not in the table below, the error message "UNDEFINED SCSI COMMAND CODE" is displayed.

SCSI CMD CODE	ROUTINE NAME	ROUTINE TITLE
08	TRECEV	RECEIVE
0A	TSEND	SEND
0B	TSEEK	SEEK

The previous routines (tasks) are described in the following paragraphs.

Receive (08_{16}) Task

Method of Execution:

1. Set WDB (40_{16}) in the Control (01_{16}) register and use the CDB (05_{16} through 07_{16}) register to set the Transfer Count (12_{16} through 14_{16}) registers.
2. Issue a Send Data (15_{16}) command.
3. Output the appropriate number of bytes of data, and wait for "Send Command Completed Successfully" (13_{16}) interrupt.
4. Issue a Send Status (14_{16}) command.
5. Set the status in the Data (19_{16}) register after Data Buffer Ready sets.
6. Wait for "Send Command Completed Successfully" (13_{16}) interrupt.
7. Issue a Send Message In (16_{16}) command.
8. Set the Command Complete (00_{16}) Message in the Data (19_{16}) register after Data Buffer Ready sets.
9. Wait for "Send Command Completed Successfully" (13_{16}) interrupt.
10. Issue a Disconnect (04_{16}) command.

Send ($0A_{16}$) Task

Method of Execution:

1. Set WDB (40_{16}) in the Control (01_{16}) register and use the CDB (05_{16} through 07_{16}) registers to set the Transfer Count (12_{16} through 14_{16}) registers.
2. Issue a Receive Data (11_{16}) command.
3. Input the appropriate number of bytes of data, and wait for "Receive Command Completed Successfully" (13_{16}) interrupt.
4. Issue a Send Status (14_{16}) command.
5. Set the status in the Data (19_{16}) register after Data Buffer Ready sets, and wait for "Send Command Completed Successfully" (13_{16}) interrupt.
6. Issue a Send Message In (16_{16}) command.
7. Set the Command Complete (00_{16}) Message in the Data (19_{16}) register after Data Buffer Ready sets, and wait for "Send Command Completed Successfully" (13_{16}) interrupt.
8. Issue a Disconnect (04_{16}) command.

Seek ($0B_{16}$) Task

Method of Execution:

1. Issue a Send Message In (16_{16}) command.
2. Set the Disconnect (04_{16}) Message in the Data (19_{16}) register after Data Buffer Ready sets, and wait for "Send Command Completed Successfully" (13_{16}) interrupt.
3. Issue a Disconnect (04_{16}) command.
4. Set WDB (40_{16}) in the Control (01_{16}) register, use the CDB (05_{16} through 07_{16}) register in order to set the Transfer Count (12_{16} through 14_{16}) registers.
5. Clear the Target LUN/Status ($0F_{16}$) register.
6. Issue a Reselect and Send Data ($0B_{16}$) command.
7. Output the appropriate number of bytes of data, and wait for "Send Command Completed Successfully" (13_{16}) interrupt.
8. Issue a Send Status (14_{16}) command.
9. Set the status in the Data (19_{16}) register after Data Buffer Ready sets, and wait for "Send Command Completed Successfully" (13_{16}) interrupt.
10. Issue a Send Message In (16_{16}) command.
11. Set the Command Complete (00_{16}) Message in the Data (19_{16}) register after Data Buffer Ready sets, and wait for "Send Command Completed Successfully" (13_{16}) interrupt.
12. Issue a Disconnect (04_{16}) command.

IP25/IP252/IP254/IP256 IPI 25-MB Channel Adapter 9

IP25/IP252/IP254/IP256 IPI 25-MB Channel Adapter 9

Section Descriptions

Section 00 – IPI Adapter PP Access Test

This section tests the IPI adapter access from the peripheral processor (PP). Channel signal response is verified by using the Master Clear Adapter function, Clear Error function, Illegal function ($0F00_{16}$), Read Error Status function with channel I/O, and the Invert PP to Channel Parity facility.

Subsection 0

Tests the response on Master Clear Adapter function.

Method of Execution:

- Issue a Master Clear Adapter (0000_{16}) function to the IPI channel adapter.

All bytes in the MAC IPI Channel Adapter Status register except for the DMA Operational Status register and IPI Status register are checked to be zero.

```
( optional error data ) none
```

Subsection 1

Tests the response on clear error function.

Method of Execution:

- Issue a Clear Error (0100_{16}) function to the IPI channel adapter.

The MAC IPI Channel Adapter Status register-s containing the DMA Error Status register and IPI Error register status bytes are checked to be zero.

```
( optional error data ) none
```

Subsection 2

Tests the response on Read DMA Error Status Register function.

Method of Execution:

1. Issue a Read DMA Error Status register (0600_{16}) function to the IPI channel adapter.
2. Read and check the DMA Error Status register for being all zeros. The Fault Status 2 register is also checked.

```
( optional error data ) none
```

Subsection 3

Tests the response on Illegal function.

Method of Execution:

1. Issue the Illegal functions (0400_{16}, 0500_{16}, and $0F00_{16}$) to the IPI channel adapter.
2. Read and check the DMA error status register for Illegal function bit set.
3. Check the MAC IPI Channel Adapter Status register's DMA Error Status register byte zero for Illegal function bits. The Fault Status 2 register is also checked.

```
( optional error data ) none
```

Subsection 4

Tests the Channel Data register parity checker.

Method of Execution:

1. Enable the maintenance channel's Invert PP to Channel Parity at Generator (IOU Test Mode Maintenance register test code 02B).
2. Place a data pattern on the channel and read.
3. Check the MAC IPI Channel Adapter Status register's Flag Status register Byte 5 for the active flag, full flag and error flag, status bits being set.
4. Read a data pattern from the channel.
5. Check the MAC IPI Channel Adapter Status register's Flag Status register Byte 5 for the error flag status bit being set.
6. Check the Fault Status 2 register for the channel error bit being set.
7. Check the MAC IPI Channel Adapter Status register's DMA Error Status register byte one and DMA Error Status register for BAS parity error and MX error status bits being set.

This subsection consists of four conditions, each being a different data pattern. This subsection is not to be executed in multiprocessing mode (see PARAM15).

```
( optional error data )
PATTERN = PPPP
```

Subsection 5

Tests the response on function parity error.

Method of Execution:

1. Enable the maintenance channel's Invert PP to Channel Parity at Generator (IOU Test Mode Maintenance register test code 02B).
2. Issue a function (0800_{16}) to the IPI adapter.
3. Check the DMA Error Status and Fault Status 2 register.
4. Check the MAC IPI Channel Adapter Status register's DMA Error Status register byte one and DMA Error Status register for BAS parity error and MX error status bits being set.

This subsection is not executed in multiprocessing mode (see PARAM15).

```
( optional error data ) none
```

Section 01 – IPI Channel Adapter Control Register Test

This section tests the IPI Channel Adapter Control register using the Write Control register and Read Control register functions and the Invert Control register parity facility.

Subsection 0

Tests the interface between the channel data register and control register.

Method of Execution:

1. Load data into the Control register using the Write Control register (0300_{16}) function.
2. Issue a Read Control register (0200_{16}) function and read data.
3. Check the MAC IPI Adapter Status register's DMA Control register Byte 6 for the appropriate bits being set.
4. Check the data read from the DMA Error Status register for errors.

This subsection consists of 17 conditions, each being a different pattern.

```
( optional error data )
CONTROL REGISTER = cccc
```

Subsection 1

Tests the control register parity generator.

Method of Execution:

1. Load data into the Control register.
2. Read the Control register and compare data.
3. Check the DMA Error Status register for errors.

This subsection consists of 256 conditions, each being a different pattern.

```
( optional error data )
CONTROL REGISTER = cccc
```

Subsection 2

Tests that Master Clear Adapter function clears the Control register.

Method of Execution:

1. Load nonzero data into the Control register.
2. Issue a Master Clear Adapter function.
3. Read the register and check data to be zero.

```
( optional error data ) none
```


Subsection 3

Tests Force BAS register parity error.

Method of Execution:

1. Set Force BAS register parity error upper (force error code 06_{16}) or Force BAS register parity error lower (force error code 07_{16}) in the Control register force error code (one per condition).
2. Output and input a word on the IPI channel.
3. Check the MAC IPI Channel Adapter Status register for the DMA Error Status register bits DMA register parity error, BAS parity error, and MX error.
4. Check Fault Status 1 for being correct.

```
( optional error data )
CONTROL REGISTER = cccc
```

Section 02 – DMA Operational Register Status Test.

This section tests the DMA enhanced IPI channel adapter's DMA Operational Status register.

Subsection 0

Tests the response to a read DMA Operational Status register function and the interface between the Operational Status register and the Channel Data register.

Method of Execution:

1. Issue a Read DMA Operational Status register (0700_{16}) function to the IPI channel adapter and the DMA Operational Status register.
2. Read and check the Operational Status register for the appropriate bits set.
3. Check the Fault Status 2 register and DMA Error Status register.

```
( optional error data ) none
```

Section 03 – T-Register Test

This section tests the DMA Enhanced IPI Channel Adapter's T register operations using the Clear T register, Read T register and Write T register functions and force parity.

Subsection 0

Tests the Write T register function.

Method of Execution:

1. Issue a Write T register ($0B00_{16}$) function and output three words.
2. Attempt to output a fourth word.
3. Check the channel signal response; Full should remain set.
4. Check the DMA Error Status register.

```
( optional error data ) none
```

Subsection 1

Tests the T Prime register empty bit and T register Data Transfer functions.

Method of Execution:

1. Write the T register.
2. Check the state of the T Prime register empty bit for being set in the DMA Operational Status register and the MAC IPI Channel Adapter Status register's DMA Operational Status register Byte 4.
3. Write the T registers.
4. Check the state of the T Prime register empty bit for being clear in the DMA Operational Status register and the MAC IPI Channel Adapter Status register's DMA Operational Status register Byte 4.
5. Read the T register compare data with the data written.

This subsection consists of seven conditions, each being a different pattern.

```
( optional error data )
T-REGISTER bbbb tttt tttt
```

Subsection 2

Tests the Clear T register function.

Method of Execution:

1. Write both T registers.
2. Check the T Prime register Empty for the clear state.
3. Clear the T registers using the Clear T Register function.
4. Read the T register and the check data for being zeros.

```
( optional error data )
T-REGISTER bbbb tttt tttt
```

Subsection 3

Tests the interface between the T registers and the Channel Data register.

Method of Execution:

1. Load the data into the T Prime register using the Write T Prime register ($0B00_{16}$) function.
2. Read the T register using the Read T register ($0A00_{16}$) function.
3. Check the data.
4. Check the DMA Error Status register for errors.

This subsection consists of 45 conditions, each being a different pattern.

```
( optional error data )
T-REGISTER bbbb tttt tttt
```

Subsection 4

Tests the T register parity checkers.

Method of Execution:

1. Master clear the IPI channel adapter.
2. Set the Force T register Data Parity Bit Low (force error code 92_{16}) in the Control register force error code.
3. Load a data pattern into the T register twice.
4. Check the DMA Error Status register for HY error.

This subsection consists of seven conditions, each being a different data pattern.

```
( optional error data )
T REGISTER bbbb tttt tttt
```

Subsection 5

Tests the T register parity generators.

Method of Execution:

1. Load the T register with a data pattern.
2. Read the contents of the T register and the check data.
3. Check the DMA Error Status register for errors.

This subsection consists of 30 conditions, each being a different pattern.

```
( optional error data )
T-REGISTER bbbb tttt tttt
```

Subsection 6

Tests the T Prime register empty status.

Method of Execution:

1. Set the T Prime register empty bit in the Control register.
2. Issue a Clear T register ($0E00_{16}$) function.
3. Clear the channel flag.
4. Load the T register with a data pattern.
5. Check the channel flag for being set and DMA Operational Status register.
6. Check the MAC IPI Channel Adapter Status register's Flag Status register Byte 5 for channel flag status bit being set.
7. Clear the channel flag.
8. Load the T register with the same data pattern.
9. Check the channel flag for being clear and DMA Operational Status register.

```
( optional error data )
T-REGISTER bbbb tttt tttt
```

Section 04 – IPI Static Force Error Code Function Test

This section tests the IPI Static Force Error Code functions. The section uses the IPI Error register status and the DMA Error Status register (IPI error bit 55) to test for the forced errors.

Subsection 0

Tests the Response On Read IPI Error Register function.

Method of Execution:

1. Issue a Read IPI Error Register ($00F1_{16}$) function to the IPI channel adapter after a Clear IPI Error (0022_{16}) function.
2. Read and check to verify that all bits are clear.
3. Check the DMA Error Status register and Fault Status 2 register.

   (optional error data) none

Subsection 1

Tests the response on Read IPI Status Register function.

Method of Execution:

1. Issue a Clear IPI Error (0022_{16}) function.
2. Issue a Read IPI Status register ($00E1_{16}$) function to the IPI channel adapter.
3. Read and check the IPI Status register to be zero.
4. Check the DMA Error Status register and Fault Status 2 register for errors.

   (optional error data) none

Subsection 2

Tests the response on Invert Input Data Parity to BAS.

Method of Execution:

1. Set the Invert Input Data Parity to BAS (force error code 08_{16}) in the Control register force error code.
2. Check the IPI Status register for being all zeros.
3. Read and check the DMA Error Status register for the BAS parity error and MX error bits set.

   (optional error data) none

Section 05 – Channel Environment Test

This section tests the IPI channel adapter environment status and select environment functions.

Subsection 0

Tests the response to the Read Environment Status function.

Method of Execution:

1. Issue a Read Environment Status ($001E_{16}$) function to the IPI channel adapter.
2. Read and check the environment status to be correct [bit 0 (DMA available/selected) set, bit 1 (IPI master channel available/selected) set, bits 2-14 (reserved) clear, and bit 15 (environment) set in the first status word and bit 8 (IPI Master Interface) in the second status word is selected].
3. Check the IPI Error and DMA Error Status registers for errors.

   (optional error data) none.

Subsection 1

Tests the response to the Select Environment function.

Method of Execution:

1. Issue a Select Environment Status ($013E_{16}$) function to the IPI channel adapter.
2. Read and check the environment status to be correct [bit 0 (DMA available/selected) set, bit 1 (IPI master channel available/selected) set, bits 2-14 (reserved) clear, and bit 15 (environment) set in the first status word and the second status word is expected to be equal to the function code].
3. Check the IPI Error and DMA Error Status registers for errors.

   (optional error data) none.

Subsection 2

Tests the response to the select environment functions that select/deselect environments.

Method of Execution:

1. Issue a Select Environment Status ($0X3E_{16}$) function to the IPI channel adapter.
2. Read and check the environment status to be correct [bit 0 (DMA available/selected) set, bit 1 (IPI master channel available/selected) set, bits 2-14 (reserved) clear, and bit 15 (environment) set in the first status word and the second status word is expected to be equal to the function code].
3. Issue a Master Clear (0000_{16}) function to the IPI channel adapter and is not expected to be replied to except for the $013E_{16}$ function code.
4. Issue a Master Clear using the IOU Master Clear from the MAC to clear the IPI channel adapter.
5. Read and check the environment status to be correct [bit 0 (DMA available/selected) set, bit 1 (IPI master channel available/selected) set, bits 2-14 (reserved) clear, and bit 15 (environment) set in the first status word and the second status word is expected to have bit 8 set].
6. Check the IPI Error and DMA Error Status registers for errors.

Yhis subsection consists of 16 conditions, each condition being a different select environment function.

```
( optional error data )
SELECT ENV. FUNCTION= xxxx
```

Section 06 – IPI Rev And IPI Option Register Test

This section tests the Read IPI Rev register and Read IPI Option register functions.

Subsection 0

Tests the response to the Read IPI Rev Register function.

Method of Execution:

1. Issue a Read IPI Rev Register (0002_{16}) function to the IPI channel adapter.
2. Read and check the IPI Rev register status to be correct (bits 14 and 15 are zero).
3. Check the DMA Error Status register for errors.

   (optional error data) none.

Subsection 1

Tests the response to the Read IPI Option register function.

Method of Execution:

1. Issue a Read IPI Option Register (0102_{16}) function to the IPI channel adapter.
2. Read and check the IPI Option register status to be correct [bit 0 (single octet mode supported) clear, bit 1 (double octet mode supported) set, bit 2 (quad octet mode supported) clear, bits 3 and 4 (reserved) clear, bits 5, (maintenance mode 1 supported) 6, (maintenance mode 2 supported), 7 (auto bus control supported), 8 (auto master/slave status supported), 9 (master status terminate supported), and 10 (port b supported), set, bits 11-13 (reserved) clear, bit 14 (stream mode supported) set, and bit 15 (interlock mode supported) set].
3. Check the DMA Error Status register for errors.

   (optional error data) none.

Section 07 – IPI Adapter Operand Register Test

This section tests the IPI Adapter Operand register and interface to the Channel data register. The section tests the IPI Adapter Write Operand register and Read Operand register functions, and checks the interface between the Operand register and the Channel Data register.

Subsection 0

Tests the reading of the Operand register and interface to the Channel Data register.

Method of Execution:

1. Issue the Master Clear (0000_{16}) function, to clear the IPI channel adapter.
2. Read the Operand register using the Read Operand Register (0702_{16}) function.
3. Check the channel error flag for errors.
4. Check the seed, burst length, and compared data words for being all zeros.

```
(optional error data)  none.
```

Subsection 1

Tests the writing and reading of the Operand register and interface from/to the Channel Data register.

Method of Execution:

1. Issue the Master Clear (0000_{16}) function, to clear the IPI channel adapter.
2. Load the Operand register using the Write Operand Register (0602_{16}) function.
3. Read the Operand register.
4. Check the channel error flag for errors.
5. Check the seed, burst length, and compared data words for being correct.

This subsection consists of 36 conditions each being a different pattern.

```
(optional error) data none.
```

Section 08 - IPI Adapter IPI Transmitter/Receiver Register Test

This section tests the IPI Transmitter/Receiver Enable register in the IPI adapter.

Method of Execution:

1. Write the IPI Transmitter/Receiver Enable register.
2. Read the data on the IPI Transmitter/Receiver Enable register.

Subsection 0

Tests the reading of the IPI Transmitter/Receiver Enable register.

Method of Execution:

1. Issue the Master Clear (0000_{16}) function, to clear the IPI channel adapter.
2. Read the IPI Transmitter/Receiver Enable register by using the read IPI Transmitter/Receiver Register (0302_{16}) function.
3. Check the channel error flag for errors.
4. Check the contents of the IPI Transmitter/Receiver Enable register for 0030_{16}.

   (optional error data) none.

Subsection 1

Tests the writing of the IPI Transmitter/Receiver Enable register.

Method of Execution:

1. Issue the Master Clear (0000_{16}) function, to clear the IPI channel adapter.
2. Write the IPI Transmitter/Receiver Enable register by using the write IPI Transmitter/Receiver Register (0202_{16}) function.
3. Read the IPI Transmitter/Receiver Enable register by using the Read IPI Transmitter/Receiver Register function.
4. Check the channel error flag for errors.
5. Check the contents of the IPI Transmitter/Receiver Enable register for being correct.
6. The Lock register until next load IPI transmitter/receiver register function bit will be forced to be set for all data patterns used in this subsection.

This subsection consists of 36 conditions each being a different pattern.

(optional error data) none.

Subsection 2

Tests the Lock register bit in the IPI Transmitter/Receiver Enable register.

Method of Execution:

1. Issue the Master Clear (0000_{16}) function, to clear the IPI channel adapter.
2. Write the IPI Transmitter/Receiver Enable register with all zeros (lock register bit clear).
3. Read the IPI Transmitter/Receiver Enable register.
4. Check status of bits 04 though 07 in the IPI Transmitter/Receiver Enable register to verify that bits 04 and 05 are set, and bits 06 and 07 are clear.
5. Check the channel error flag for errors.

   (optional error data) none.

Section 09 – IPI Channel Clock Select Function Test

This section tests the IPI channel adapter response to the functions for Attention Present ($00C1_{16}$), External Clock/Port Select ($0X62_{16}$) on the IPI channel and Set Sync Period ($XX42_{16}$) on the IPI channel.

Subsection 0

Tests the IPI channel adapter response to the Attention Present function.

Method of Execution:

1. Issue a Master Clear (0000_{16}) function, to clear the IPI channel adapter.
2. Issue the Attention Present ($00C1_{16}$) function to the IPI channel adapter.
3. Check the channel for being inactive. No attempt is made to read the IPI status that should be returned as a result of issuing this function to the IPI channel adapter.
4. Check the DMA Error Status register and IPI Error register status for errors.

   (optional error data) none.

Subsection 1

Tests the IPI channel adapter response to the set sync period on the IPI Channel functions.

Method of Execution:

1. Issue the Master Clear (0000_{16}) function, to clear the IPI channel adapter.
2. Issue one of the Set Sync Period ($XX42_{16}$) functions listed below to the IPI channel adapter.
3. Check the channel for being inactive.
4. Read the IPI Transmitter/Receiver Enable register.
5. Check the register status to verify that the correct clock is selected, the correct port is selected and that the external clock select bit is correct.
6. Check the DMA Error Status register and IPI Error register status for errors.

This subsection consists of 27 conditions, each being a different set sync period function.

Function (Hex)	20-MHz Internal Clock
0042	0.16 Mbyte
6842	0.80 Mbyte
6C42	1 Mbyte
7442	1.70 Mbyte
7642	2 Mbyte
7942	3 Mbyte
7A42	4 Mbyte
7C42	5 Mbyte
7D42	6 Mbyte
7E42	10 Mbyte

Function (Hex)	25-MHz Ext. Clk	20-MHz Ext. Clk	16-MHz Ext. Clk	12-MHz Ext. Clk
8042	0.20 Mbyte	0.16 Mbyte	0.13 Mbyte	0.09 Mbyte
E742	1 Mbyte	0.80 Mbyte	0.64 Mbyte	0.48 Mbyte
E842	1.04 Mbyte	0.83 Mbyte	0.67 Mbyte	0.50 Mbyte
EC42	1.25 Mbyte	1 Mbyte	0.80 Mbyte	0.60 Mbyte
F342	1.92 Mbyte	1.54 Mbyte	1.23 Mbyte	0.92 Mbyte
F442	2.08 Mbyte	1.67 Mbyte	1.33 Mbyte	1 Mbyte
F542	2.27 Mbyte	1.82 Mbyte	1.45 Mbyte	1.09 Mbyte
F642	2.50 Mbyte	2 Mbyte	1.60 Mbyte	1.20 Mbyte
F742	2.78 Mbyte	2.22 Mbyte	1.78 Mbyte	1.33 Mbyte
F842	3.13 Mbyte	2.50 Mbyte	2 Mbyte	1.50 Mbyte
F942	3.57 Mbyte	2.86 Mbyte	2.29 Mbyte	1.71 Mbyte
FA42	4.17 Mbyte	3.33 Mbyte	2.67 Mbyte	2 Mbyte
FB42	5 Mbyte	4 Mbyte	3.20 Mbyte	2.40 Mbyte
FC42	6.25 Mbyte	5 Mbyte	4 Mbyte	3 Mbyte
FD42	8.33 Mbyte	6.67 Mbyte	5.33 Mbyte	4 Mbyte
FE42	12.5 Mbyte	10 Mbyte	8 Mbyte	6 Mbyte
FF42	25 Mbyte	20 Mbyte	16 Mbyte	12 Mbyte

```
( optional error data )  FUNCTION CODE=FFFF
```

Subsection 2

Tests the IPI channel adapter response to the External Clock/Port Select on the IPI Channel functions.

Method of Execution:

1. Issue the Master Clear (0000_{16}) function to clear the IPI adapter.
2. Select the external clock by using the Set Sync Period Counter function.
3. Issue one of the Set External Clock ($XX62_{16}$) functions listed below to the IPI channel adapter.
4. Check the channel for being inactive.
5. Read and check the IPI Transmitter/Receiver Enable register to verify that the correct clock is selected, the correct port is selected and that the external clock select bit is correct.
6. Read and check the IPI Status register to verify that the Port B status bit is correct.
7. Check the status of the DMA Error Status register and IPI Error register for errors for only the functions that have a clock defined.

This subsection consists of 16 conditions, each being a different set external clock function.

Function	Clock	Port
0062	20 MHz	A
0162	12 MHz	A
0262	16 MHz	A
0362	25 MHz	A
0462	TBD	A
0562	TBD	A
0662	TBD	A
0762	TBD	A
0862	20 MHz	B
0962	12 MHz	B
0A62	16 MHz	B
0B62	25 MHz	B
0C62	TBD	B
0D62	TBD	B
0E62	TBD	B
0F62	TBD	B

```
( optional error data )  FUNCTION CODE=FFFF
```

Section 10 – IPI Adapter Output Register/Bus Input Test.

This section tests the Write IPI Output register and Read IPI Input Bus in the IPI adapter. The section tests Bus A and Bus B by placing data on the buses using the Write IPI Output Register function and reads the data from Bus A and Bus B using the Read IPI Bus Input function.

Subsection 0

Tests the writing of the IPI Output register for Bus A and Bus B.

Method of Execution:

1. Issue the Master Clear (0000_{16}) function, to clear the IPI channel adapter.
2. Enable the Bus A and Bus B transmitters and receivers using the Write IPI Transmitter/Receiver Register function.
3. Write the Output register for Bus A and Bus B using the Write IPI Output Register (0502_{16}) function.
4. Read the data on Bus A and Bus B by using the Read IPI Bus Input function.
5. Check the channel error flag for errors.
6. Check the Bus A and Bus B data word for being correct.

This subsection consists of 36 conditions each being a different pattern.

```
< optional error data none.
```

Section 11 – IPI Bus Status And Control Test

This section tests the IPI Bus Control functions, Bus B status, and IPI Status register. The section uses the Bus Control function ($XXX1_{16}$), its Bus B status response and the Read IPI Status Register function ($00E0_{16}$) to check the operation of the IPI control lines, select out, master out, and sync out.

Subsection 0

Tests the status returned on Bus B.

Method of Execution:

1. Issue the Master Clear (0000_{16}) function, to clear the IPI channel adapter.
2. Write the IPI Transmitter/Receiver Enable register to enable the Bus A and Bus B Transmitters/Receivers and lock the register until the next IPI Transmitter/Receiver Register function.
3. Issue the Bus Control function, 0011_{16} (that is, sync out clear, master out clear, select out clear, and condition code 1 set), and the channel activated in order to read the IPI Bus B status word.
4. Check the IPI Bus B status for being equal to 0011_{16}.
5. Check the IPI Status register for being equal to 0000_{16}.
6. Check the MAC IPI Channel Adapter Status Register's IPI Status register Byte 7 for being all zeros.
7. Check the IPI Error register status and DMA Error Status register for errors.

   (optional error data) none.

Subsection 1

Tests the IPI master reset sequence and sync out signal.

Method of Execution:

1. Issue the Bus Control function, 0013_{16}.
2. Set Sync Out, clear Master Out, clear Select Out, and set Condition Code 1 is issued and the channel activated in order to read the IPI Bus B status word.
3. Check the IPI Bus B status and IPI Status register for Sync Out set and all other bits clear.
4. Check the MAC IPI Channel Adapter Status register's IPI Status register Byte 7 is for Sync Out status.
5. Issue, at least 12 microseconds later the Bus Control function, 0011_{16} with all control bits clear and Condition Code 1 set.
6. Check the IPI Bus B status and IPI Status register for all bits clear.
7. Check the IPI Error register status and DMA Error Status register for errors.

   (optional error data) none.

Subsection 2

Tests the select/deselect bus slave sequence.

Method of Execution:

1. Execute the Master Reset sequence with Condition Code 0.
2. Issue the Bus Control function, $XX29_{16}$ with Select Out and Condition Code 2 set, in test mode and the channel activated in order to read the IPI Bus B status word.
3. Check the IPI Bus B status and IPI Status register for Select Out and Slave In set and all other bits clear.
4. Check the MAC IPI Channel Adapter Status register's IPI Status register Byte 7 for Select Out and Slave In status.
5. Issue the Bus Control function, $XX71_{16}$ with Select Out clear and Condition Code 7 set.
6. Check the IPI Bus B status and IPI Status register for Select Out and Slave In cleared.
7. Check the IPI Error register status and DMA Error register status for errors.

   (optional error data) none.

Subsection 3

Test Condition Code 3 on Select/Deselect.

Method of Execution:

1. Execute the Master Reset sequence with Condition Code 1.
2. Issue the Bus Control function, $XX39_{16}$ with Select Out and Condition Code 3 set in test mode and the channel activated in order to read the IPI Bus B status word.
3. Check the IPI Bus B status and IPI Status register for Select Out and Slave In set and all other bits clear.
4. Check the MAC IPI channel adapter status register's IPI Status register Byte 7 for Select Out and Slave In status.
5. Issue the Bus Control function, $XX71_{16}$ (that is, with Select Out clear and Condition Code 7 set).
6. Check the IPI Bus B status and IPI Status register for Select Out and Slave In cleared.
7. Check the IPI Error register status and DMA Error Status register for errors.

   (optional error data) none.

Subsection 4

Tests the Master Out signal.

Method of Execution:

1. Execute the Master Reset sequence with Condition Code 1.
2. Enable Test mode in the IPI channel adapter Control register.
3. Issue the Bus Control function, $XX15_{16}$ with Master Out and Condition Code 1 set.
4. Activate the channel in order to read the IPI Bus B status word.
5. Check the IPI Bus B status and IPI Status register for Master Out set and all other bits clear.
6. Check the MAC IPI channel adapter Status register's IPI Status register Byte 7 for Master Out status.
7. Issue the Bus Control function, $XX11_{16}$ with Select Out clear and Condition Code 1 set.
8. Check the IPI Bus B status and IPI Status register for Select Out clear.
9. Check the IPI Error register status and DMA Error Status register for errors.

   (optional error data) none.

Subsection 5

Tests the Sync Out signal.

Method of Execution:

1. Execute the Master Reset sequence with Condition Code 1.
2. Enable Test mode in the IPI channel adapter Control register.
3. Issue the Bus Control function, $XX29_{16}$ with Select Out and Condition Code 2 set.
4. Activate the channel in order to read the IPI Bus B status word.
5. Check the IPI Bus B status and IPI Status register for Select Out and Slave In set with all other bits clear.
6. Check the MAC IPI channel adapter Status register's IPI Status register Byte 7 for Select Out and Slave In status.
7. Issue the Bus Control function, $XX5B_{16}$ with Select Out and Sync Out with Condition Code 5 set.
8. Activate the channel in order to read the IPI Bus B status word.
9. Check the IPI Bus B status and IPI Status register for Select Out, Sync In and Sync Out set with all other bits clear.
10. Check the MAC IPI channel adapter Status register's IPI Status register Byte 7 for Select Out, Sync Out, and Slave In status.
11. Issue the Bus Control function, $XX69_{16}$ with Select Out with Condition Code 6 set.
12. Activate the channel in order to read the IPI Bus B status word.
13. Check the IPI Bus B status and IPI Status register for Select Out set with all other bits clear.
14. Check the MAC IPI channel adapter Status register's IPI Status register Byte 7 for Select Out status.
15. Issue the Bus Control function, $XX01_{16}$ with Select Out, Master Out, and Sync Out clear with Condition Code 0 set.
16. Check the IPI Error register status and DMA Error Status register for errors.

    (optional error data) none.

Subsection 6

Tests the Condition Code 4.

Method of Execution:

1. Issue the Bus Control function, 0041_{16} with Sync Out clear, Master Out clear, Select Out clear, and Condition Code 41 set.
2. Activate the channel in order to read the IPI Bus B status word.
3. Check the IPI Bus B status, IPI Status register, IPI Error register status and DMA Error register status for all bits clear.

(optional error data) none.

Subsection 7

Tests the force ICI Output Data Parity Lower and Upper.

Method of Execution:

1. Execute the Master Reset sequence with Condition Code 1.
2. Issue the Force ICI Output Data Parity Lower function ($0C22_{16}$) to the IPI channel adapter.
3. Enable Test mode in the IPI channel adapter Control register.
4. Issue the Bus Control Function, $XX29_{16}$ with Select Out and Condition Code 2 set.
5. Check the IPI Bus B status for Select Out and Slave In set and all other bits clear.
6. Check the IPI Status register for error, Select Out and Slave In set and all other bits clear.
7. Check the IPI Error register status and DMA Error Status register for errors.
8. Check the DMA Error Status register is for IPI error and HZ error bits set.
9. Issue a Clear IPI Error function (0022_{16}) to the IPI channel adapter.
10. Check the DMA Error Status register for IPI error and HZ error bits set.
11. Issue the Bus Control function, $XX01_{16}$ with Select Out, Master Out and Sync Out, clear with Condition Code 0 set.
12. Check the IPI Error register status and DMA Error Status register for errors.

The subsection is repeated for force ICI Output Parity Upper ($0D22_{16}$).

(optional error data) none.

Subsection 8

Tests the force ICI Input Data Parity Lower and Upper.

Method of Execution:

1. Execute the Master Reset sequence with Condition Code 1.
2. Issue the Force ICI Input Data Parity Lower function ($0E22_{16}$) to the IPI channel adapter.
3. Enable Test mode in the IPI channel adapter Control register.
4. Issue the Bus Control function, $XX29_{16}$ with Select Out and Condition Code 2 set.
5. Check the IPI Bus B status for error, Select Out and IPINFER set and all other bits clear.
6. Check the IPI Error register status and DMA Error Status register for errors.
7. Check the IPI Error register status for errors.
8. Issue a Clear IPI Error function (0022_{16}) to the IPI channel adapter.
9. Check the DMA Error Status register for IPI error and HZ error bits set.
10. Issue the Bus Control function, $XX01_{16}$ with Select Out, Master Out, and Sync Out clear with Condition Code 0 set.
11. Check the IPI Error register status and DMA Error Status register for errors.

The subsection is repeated for Force ICI Input Parity Upper ($0F22_{16}$).

```
( optional error data ) none.
```

Subsection 9

Tests the IPI Illegal sequence status bit.

Method of Execution:

1. Execute the Master Reset sequence with Condition Code 1.
2. Enable Test mode in the IPI channel adapter Control register.
3. Issue the Bus Control function, $XX15_{16}$ with Master Out and Condition Code 1 set.
4. Check the IPI Bus B status for Master Out set and all other bits clear.
5. Check the IPI Error register status for errors.
6. Issue the Bus Control function, $XX1D_{16}$ with Select Out, Master Out, and Condition Code 1 set.
7. Check the IPI Bus B status for error, Select Out, Master Out and IPINFER set and all other bits clear.
8. Check the IPI Error register status and DMA Error Status register for errors.
9. Check the IPI Error register status for errors.

(optional error data) none.

Section 12 – IPI Channel Adapter Compare Unit Test

This section tests the IPI channel adapter Compare unit using the Random Data Generator Test mode facility in PP Output mode.

Subsection 0

Tests the IPI channel Transfer in Progress and PP/IPI mode status bits in the Operational Status register.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
4. Initialize the Random Data Generator with a test seed and burst count of two.
5. Issue the Bus Control function 0009_{16} to Set Select Out.
6. Initiate a PP to IPI channel adapter interlocked mode data transfer in double octet mode.
7. Place a data word (test seed) in the PPBUF data buffer and output to the IPI channel adapter.
8. Check the PP/IPI mode, T Prime Register Empty, and Transfer In Progress status bits to ensure that they are set in the MAC IPI channel adapter Status register's Operational Status register Byte 4.
9. Output another data word to the IPI channel adapter and to terminate the transfer.
10. Check the IPI Error register status and DMA Error Status register for errors.

    (optional error data) none.

Section 13 – IPI Adapter Random Data Generator to PP Test

This section tests the IPI channel adapter Random Data Generator to channel data register interface. The section also tests lost data status in the IPI Error register status and buffer not empty status in the IPI Status register.

Subsection 0

Tests the Random Data Generator to channel data register interface using the IPI input buffer via one word inputs.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a test seed and burst count of one.
4. Enable and lock all transmitters and receivers and enable the Random Data Generator.
5. Issue the Bus Control function 0009_{16} to Set Select Out.
6. Initiate a IPI channel adapter to PP interlocked mode data transfer in double octet mode.
7. Input one data word from the IPI channel adapter.
8. Check, upon completion of the data transfer, the IPI Error register status and DMA Error Status register for errors.
9. Check data input from the IPI channel adapter for errors.
10. Read and compare the Operand registers with the expected results.

This subsection consists of 36 conditions, each being a different data pattern.

```
( optional error data )
TEST SEED ssss
XFER blk   cccc bbbb tttt tttt
```

Subsection 1

Tests the Random Data Generator to Channel register via block transfers.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a test seed and burst count of 256 words.
4. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
5. Initiate a IPI channel adapter to PP interlocked mode data transfer in double octet mode.
6. Input 256 words from the IPI channel adapter.
7. Check, after completion of the data transfer, the IPI Error register status and DMA Error Status register for errors.
8. Read and compare the Operand registers with the expected results.
9. Check data input from the IPI channel adapter for errors.

```
( optional error data )
 TEST SEED ssss
 XFER blk bbbb tttt tttt
```

Subsection 2

Tests force lost data error status.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a test seed and burst count of 257 words.
4. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
5. Initiate a IPI channel adapter to PP interlocked mode data transfer in double octet mode.
6. Input 256 words from the IPI channel adapter with one word remaining in the IPI channel adapter.

7. Check, upon completion of the data transfer, the IPI Error register status and DMA Error Status register for errors.

8. Check data input from the IPI channel adapter for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```


Subsection 3

Tests force buffer not empty status.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.

2. Execute the IPI Master Reset sequence on Port A.

3. Initialize the Random Data Generator with a test seed and burst count of 256 words.

4. Enable and lock all transmitters and receivers, and enable the Random Data Generator.

5. Initiate a IPI channel adapter to PP interlocked mode data transfer in double octet mode.

6. Input 257 words from the IPI channel adapter with three words remaining in the IPI channel adapter.

7. Check, upon completion of the data transfer, the IPI Status register, IPI Error register status and DMA Error Status register for errors.

```
( optional error data )
TEST seed ssss
XFER blk bbbb tttt tttt
```

Section 14 - Deadman Timeout Test

This section tests the deadman timer, the disable transfer time-out control, and time-out status.

Subsection 0

Tests deadman time-out.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a test seed and burst count of four.
4. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
5. Issue the Bus Control function 0009_{16} is issued to Set Select Out.
6. Initiate a PP to IPI channel adapter interlocked mode data transfer in double octet mode.
7. Output a data word (test seed) to the IPI channel adapter and the program waits for 1.2 seconds for the deadman timer to expire and set the channel error flag.
8. Check the MAC IPI channel adapter Status register's DMA Error Status register Byte 1 for time-out status.
9. Check Fault Status 2 register for errors.
10. Check the IPI Error register status and the DMA Error Status register for errors.

```
( optional error data ) none.
```

Subsection 1

Tests inhibit deadman timeout.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a test seed and burst count of four.
4. Enable disable transfer time-out in the DMA Control register.
5. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
6. Issue the Bus Control function 0009_{16} to Set Select Out.
7. Initiate a PP to IPI channel adapter interlocked mode data transfer in double octet mode.
8. Output a data word (test seed) to the IPI channel adapter and the program waits for 1.2 seconds for the deadman timer to expire and set the channel error flag.
9. Check, after 1.2 seconds has expired, the MAC IPI channel adapter Status register's DMA Error Status register Byte 1 for errors.
10. Check Fault Status 2 register for errors.
11. Check the IPI Error register status and the DMA Error Status register for errors.

```
( optional error data ) none.
```

Section 15 – PP Double Octet Mode Transfer Test

This section is the PP transfer test. Data is transferred between the PP and the Random Data Generator in the IPI channel adapter and between the IPI channel adapter and PP in double octet mode.

Subsection 0

Tests IPI channel adapter to PP interlocked mode data transfers in double octet mode.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a test seed and burst count of 200.
4. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
5. Issue the Bus Control function 0009_{16} to Set Select Out.
6. Initiate a IPI channel adapter to the PP interlocked mode data transfer in double octet mode.
7. Input 200 words from the IPI channel adapter.
8. Check, upon completion of the data transfer, the IPI Error register status and DMA Error Status register for errors.
9. Check data input from the IPI channel adapter for errors.

This subsection consists of 10 conditions, each being a different random data pattern.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 1

Tests PP to IPI channel adapter interlocked mode data transfers in double octet mode.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a test seed and burst count of 200.

4. Enable and lock all transmitters and receivers and enable the Random Data Generator.
5. Issue the Bus Control function 0009_{16} to Set Select Out.
6. Initiate a PP to IPI channel adapter interlocked mode data transfer in double octet mode.
7. Output 200 words are to the IPI channel adapter.
8. Check, upon completion of the data transfer, the IPI Error register status and DMA Error Status register for errors.

This subsection consists of 10 conditions, each being a different random data pattern.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 2

Tests IPI channel adapter to PP streaming mode transfers in double octet mode.

Method of Execution:

Same as subsection 0, only streaming mode is used.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 3

Tests PP to IPI channel adapter streaming mode transfers in double octet mode.

Method of Execution:

Same as subsection 1, only streaming mode is used.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 4

Tests force IPI bus A and B output parity.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a test seed and burst count of 200.
4. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
5. Issue the Force IPI Bus A Output Parity (0122_{16}) function to the IPI channel adapter.
6. Issue the Bus Control function 0009_{16} to Set Select Out.
7. Initiate a PP to IPI channel adapter interlocked mode data transfer in double octet mode.
8. Output 200 words to the IPI channel adapter.
9. Check, upon completion of the data transfer, the IPI Error register status and DMA Error Status register for errors.

The subsection is repeated for Force IPI Bus B Output Parity (0222_{16}).

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 5

Tests force IPI Bus A and B Input Parity.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a test seed and burst count of 200.
4. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
5. Issue the Force IPI Bus A Input Parity (0322_{16}) function to the IPI channel adapter.
6. Issue the Bus Control function 0009_{16} to Set Select Out.
7. Initiate a PP to IPI channel adapter interlocked mode data transfer in double octet mode.
8. Output 200 words to the IPI channel adapter.
9. Check, upon completion of the data transfer, the IPI Error register status and DMA Error Status register for errors.

The subsection is repeated for Force IPI Bus B Input Parity (0422_{16}).

```
( optional error data )
TEST seed ssss
XFER blk bbbb tttt tttt
```

Section 16 – Transfer Complete Test

This section tests setting of transfer complete at the end of a PP mode transfer.

Subsection 0

Tests that transfer complete doesn't set the channel flag in PP mode.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a test seed and burst count of one.
4. Enable the Enable Tip Flag in the IPI channel adapter control register.
5. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
6. Issue the Bus Control function 0009_{16} to Set Select Out.
7. Initiate a PP to IPI channel adapter interlocked mode data transfer in double octet mode.
8. Output one data word to the IPI channel adapter.
9. Check, upon completion of the data transfer, the IPI Error register status and DMA Error Status register for errors.
10. Check the channel flag for being clear.

```
( optional error data )
TEST seed ssss
XFER blk bbbb tttt tttt
```

Section 17 – Central Memory to IPI Transfer Register Interface Test

This section tests the central memory to IPI Transfer register interface using the test mode facility in DMA read mode and invert parity.

Subsection 0

Tests transfer complete setting of channel flag.

Method of Execution:

1. Initialize central memory with a PP unique data pattern.
2. set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
3. Execute the IPI Master Reset sequence on Port A.
4. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
5. Enable the Enable Tip Flag in the adapter DMA Control register.
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue the Bus Control function 0009_{16} to Set Select Out.
8. Load the T-register with the central memory address and byte count of eight (one central memory word).
9. Initiate a central memory to IPI channel adapter interlocked mode data transfer in double octet mode.
10. Check, upon completion of the data transfer, the IPI Error register status and DMA Error Status register for errors.
11. Check the channel flag for being set.

```
( optional error data )
TEST seed ssss
XFER blk bbbb tttt tttt
```

Subsection 1

Tests the disassembly buffer ranks. This subsection is bypassed if PARAM15 is set.

Method of Execution:

1. Initialize central memory with a PP unique data pattern.
2. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
3. Execute the IPI Master Reset sequence on Port A.
4. Initialize the Random Data Generator with a data pattern and burst count of four.
5. Enable the Enable Tip Flag in the adapter DMA Control register.
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator. Central memory requests are blocked at the CMI.
7. Issue the Bus Control function 0009_{16} to Set Select Out.
8. Load the T-register with the central memory address and byte count of eight (one central memory word).
9. Initiate a central memory to IPI channel adapter interlocked mode data transfer in double octet mode.
10. Release central memory requests at the CMI to allow the transfer to complete.
11. Check, upon completion of the data transfer, the IPI Error register status and DMA Error Status register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 2

Tests the disassembly buffer.

Method of Execution:

1. Initialize central memory with a PP unique data pattern.
2. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
3. Execute the IPI Master Reset sequence on Port A.
4. Initialize the Random Data Generator with a data pattern and burst count of four.
5. Enable the Enable Tip Flag in the adapter DMA Control register.
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue the Bus Control function 0009_{16} to Set Select Out.

8. Load the T register with the central memory address and byte count of eight (one central memory word).
9. Initiate a central memory to IPI channel adapter interlocked mode data transfer in double octet mode.
10. Check, upon completion of the data transfer, the IPI Error register status, DMA Error Status register and Fault Status 1 register for errors.

This subsection consists of 36 conditions, each being a different pattern. The real memory address and byte count remain unchanged for each condition.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```


Subsection 3

Tests the disassembly buffer parity generators.

Method of Execution:

1. Initialize central memory with a PP unique data pattern.
2. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
3. Execute the IPI Master Reset sequence on Port A.
4. Initialize the Random Data Generator with a data pattern and burst count of four.
5. Enable the Enable Tip Flag in the adapter DMA Control register.
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue the Bus Control function 0009_{16} to Set Select Out.
8. Load the T register with the central memory address and byte count of eight (one central memory word).
9. Initiate a central memory to IPI channel adapter interlocked mode data transfer in double octet mode.
10. Check, upon completion of the data transfer, the IPI Error register status, DMA Error Status register and Fault Status 1 register.
11. Read and check the test seed for errors.

This subsection consists of 32 conditions, each being a different pattern. The real memory address and byte count remain unchanged for each condition.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 4

Tests the assembly buffer lower byte parity checker.

Method of Execution:

1. Initialize central memory with a PP unique data pattern.
2. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
3. Execute the IPI Master Reset sequence on Port A.
4. Initialize the Random Data Generator with a data pattern and burst count of four.
5. Enable the Enable Tip Flag and the DMA Control register force error code for invert lower channel output parity (force error code 14_{16}) in the IPI adapter DMA Control register.
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue the Bus Control function 0009_{16} to Set Select Out.
8. Load the T register with the central memory address and byte count of eight (one central memory word).
9. Initiate a central memory to IPI channel adapter interlocked mode data transfer in double octet mode.
10. Check, upon completion of the data transfer, the IPI Error register status, DMA Error Status register and Fault Status 1 register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 5

Tests the assembly buffer upper byte parity checker.

Method of Execution:

Same as subsection 4 only invert upper channel output parity (force error code 13_{16}) is set in the IPI adapter DMA Control register force error code.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Section 18 – IPI To Central Memory Interface Test

This section tests the IPI Transfer register to central memory interface utilizing the test mode facility in DMA write mode and invert parity.

Subsection 0

Tests transfer complete setting of channel flag.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Tip Flag in the adapter DMA Control register.
5. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
6. Issue the Bus Control function 0009_{16} to Set Select Out.
7. Load the T register with the central memory address and byte count of eight (one CM word).
8. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
9. Check, upon completion of the data transfer, the channel flag for being set and the DMA Error Status register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 1

Tests the IPI Transfer register to central memory interface.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Tip Flag in the adapter DMA Control register.
5. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
6. Issue the Bus Control function 0009_{16} to Set Select Out.
7. Load the T register with the central memory address and byte count of eight (one central memory word).
8. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
9. Check, upon completion of the data transfer, the DMA Error Status register and Fault Status 1 register for errors.
10. Check data in central memory for errors.

This subsection consists of 36 conditions, each being a different pattern. The real memory address and byte count remain unchanged for each condition.

```
optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 2

Tests the assembly buffer ranks. This subsection is bypassed if PARAM15 is set.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Tip Flag in the adapter DMA Control register.
5. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
6. Block central memory requests at the CMI.
7. Issue the Bus Control function 0009_{16} to Set Select Out.
8. Load the T register with the central memory address and byte count of eight (one central memory word).
9. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
10. Release central memory requests at the CMI to allow the transfer to complete.
11. Check, upon completion of the data transfer, the DMA Error Status register and Fault Status 1 register.
12. Check data in central memory for errors.

This subsection consists of 36 conditions, each being a different pattern. The real memory address and byte count remain unchanged for each condition.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 3

Tests the assembly buffer parity generators.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Tip Flag in the adapter DMA Control register.
5. Enable and lock all transmitters and receivers, and the Random Data Generator.
6. Issue the Bus Control function 0009_{16} to Set Select Out.
7. Load the T register with the central memory address and byte count of eight (one central memory word).
8. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
9. Check, upon completion of the data transfer, the DMA Error Status register and Fault Status 1 register for errors.
10. Check data in central memory for errors.

This subsection consists of four conditions, each being a different pattern. The real memory address and byte count remain unchanged for each condition.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 4

Tests the assembly buffer lower byte parity checker.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Tip Flag and the DMA Control register Error Force Code for assembly buffer lower byte parity (force error code 11_{16}) in the IPI adapter DMA Control register.
5. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
6. Issue the Bus Control function 0009_{16} to Set Select Out.
7. Load the T register with the central memory address and byte count of eight (one central memory word).
8. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
9. Check, upon completion of the data transfer, the channel flag for being clear, the DMA Error Status register and Fault Status 1 register for errors.

This subsection consists of three conditions, each being a different pattern. The real memory address and byte count remain unchanged for each condition.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 5

Tests the assembly buffer upper byte parity checker.

Method of Execution:

Same as subsection 4 only force assembly buffer upper byte parity (force error code 10_{16}) is set in the IPI adapter DMA Control register force error code.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Section 19 – DMA Transfer Register Address Incrementer Test

This section tests the DMA Transfer register address incrementer utilizing the Test mode facility in DMA Read mode.

Subsection 0

Tests the DMA Transfer register address incrementer.

Method of Execution:

1. Initialize the central memory address that will be read if the address is within the defined central memory buffer boundaries.
2. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
3. Execute the IPI Master Reset sequence on Port A.
4. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
5. Enable the Enable Tip Flag in the adapter DMA Control register.
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue the Bus Control function 0009_{16} to Set Select Out.
8. Load the T register with the central memory address and byte count of eight (one central memory word).
9. Initiate a central memory to IPI channel adapter interlocked mode data transfer in double octet mode.
10. Check, upon completion of the data transfer, the DMA Error Status register and Fault Status 2 register for errors.
11. Read and check check the T register for errors.

This subsection consists of 56 conditions, each being a different central memory address pattern.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 1

Tests force parity prediction error.

Method of Execution:

1. Initialize central memory with a zeros data pattern.
2. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
3. Execute the IPI Master Reset sequence on Port A.
4. Initialize the Random Data Generator with a all ones data pattern and burst count of four.
5. Enable the Enable Tip Flag and the DMA Control register force error code for force address parity prediction error (force error code 15_{16}) in the IPI adapter DMA Control register.
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue the Bus Control function 0009_{16} to Set Select Out.
8. Load the T register with the central memory address and byte count of eight (one central memory word).
9. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
10. Check, upon completion of the data transfer, the DMA Error Status register and Fault Status 1 register for errors.
11. Check data in central memory for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Section 20 – Enable Double CMI Slot Test

This section tests the Enable Double CMI slot bit in the DMA Control register in the IPI channel adapter. The section is bypassed if the IPI 25-MB DMA channel adapter being tested cannot be tested at the 25-MB transfer rate.

Subsection 0

Tests the enable double CMI slot bit in the DMA Control register during a IPI to central memory data transfer.

Method of Execution:

1. Initialize central memory with a data pattern of all zeros.
2. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
3. Execute the IPI Master Reset sequence on Port A.
4. Initialize the Random Data Generator with a PP unique data pattern and burst count of 64.
5. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue the Bus Control function 0009_{16} to Set Select Out.
8. Load the T register with the central memory address and byte count of 128 (16 central memory words).
9. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
10. Check, upon completion of the data transfer, the DMA Error Status register for errors.
11. Check data in central memory for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 1

Tests the Enable Double CMI slot bit in the DMA Control register during a central memory IPI data transfer.

Method of Execution:

1. Initialize central memory with the data pattern.
2. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
3. Execute the IPI Master Reset sequence on Port A.
4. Initialize the Random Data Generator with a PP unique data pattern and burst count of 64.
5. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue the Bus Control function 0009_{16} to Set Select Out.
8. Load the T register with the central memory address and Byte count of 128 (16 central memory words).
9. Initiate a central memory to IPI channel adapter interlocked mode data transfer in double octet mode.
10. Check, upon completion of the data transfer, the DMA Error Status register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Section 21 – Internal Clock Transfer Test

This section tests the data transfers using the internal clock in the IPI channel adapter.

Subsection 0

Tests port A internal clock transfer test.

Method of Execution:

1. Set the sync period to to the 25-MHz internal clock and the transfer rate set to one of the transfer rates as determine by the number of the condition.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a random data pattern and burst count of 64.
4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
6. Issue the Bus Control function 0009_{16} to Set Select Out.
7. Load the T register with the central memory address and byte count of 128 (16 central memory words).
8. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
9. Check, upon completion of the data transfer, the DMA Error Status register for errors.
10. Set the sync period to to the 25-MHz internal clock and the transfer rate set to one of the transfer rates as determine by the number of the condition.
11. Execute the IPI Master Reset sequence on Port A.
12. Initialize the Random Data Generator with a random data pattern and burst count of 64.
13. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
14. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
15. Issue the Bus Control function 0009_{16} to Set Select Out.
16. Load the T register with the central memory address and byte count of 128 (16 central memory words).

17. Initiate a central memory to IPI channel adapter interlocked mode data transfer in double octet mode.

18. Check, upon completion of the data transfer, the DMA Error Status register for errors.

This subsection consists of 10 conditions, each being a different random data pattern.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```


Subsection 1

Tests Port B internal clock transfer test.

Method of Execution:

Same as Subsection 0, but Port B is used instead of Port A.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Section 22 – Port A External Clock Test

This section tests the data transfers using the external clock and Port A in the IPI channel adapter.

Subsection 0

Tests transfers using the 25-MHz external clock on Port A.

Method of Execution:

1. Set the sync period to the 25-MHz external clock and the transfer rate set to one of the transfer rates as determined by the number of the condition.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a random data pattern and burst count of 64.
4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
6. Issue the Bus Control function 0009_{16} to Set Select Out.
7. Load the T register with the central memory address and byte count of 128 (16 central memory words).
8. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
9. Check, upon completion of the data transfer, the DMA Error Status register for errors.

This subsection consists of 16 conditions, each being a different transfer rate.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 1

Tests transfers using the 16-MHz external clock on Port A.

Method of Execution:

Same as Subsection 0, but the 16-MHz external clock on Port A is used.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```


Subsection 2

Tests transfers using the 12-MHz external clock on Port A.

Method of Execution:

Same as Subsection 0, but the 12-MHz external clock on Port A is used.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Section 23 – Port B External Clock Test

This section tests the data transfers using the external clock and Port B in the IPI channel adapter.

Subsection 0

Tests transfers using the 25 MHz external clock on Port B.

Method of Execution:

1. Set the sync period to the 25 MHz external clock and the transfer rate set to one of the transfer rates as determined by the number of the condition.
2. Execute the IPI Master Reset sequence on Port B.
3. Initialize the Random Data Generator with a random data pattern and burst count of 64.
4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
6. Issue the Bus Control function 0009_{16} to Set Select Out.
7. Load the T register with the central memory address and byte count of 128 (16 central memory words).
8. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
9. Check, upon completion of the data transfer, the DMA Error Status register for errors.

This subsection consists of 16 conditions, each being a different transfer rate.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 1

Tests transfers using the 16-MHz external clock on Port B.

Method of Execution:

Same as Subsection 0, but the 16-MHz external clock on Port B is used.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 2

Tests transfers using the 12-MHz external clock on Port B.

Method of Execution:

Same as Subsection 0, but the 12-MHz external clock on Port B is used.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Section 24 – DMA Transfer Register Byte Count Test

This section tests the DMA Transfer register byte counter decrementer using the Test Mode facility in DMA Write mode.

Subsection 0

Tests the DMA Transfer register byte counter decrementer.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of maximum number of central memory words available multiplied by four.
4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
6. Issue the Bus Control function 0009_{16} to Set Select Out.
7. Load the T register with the central memory address and byte count for the maximum number of central memory words available.
8. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
9. Check, upon completion of the data transfer, the DMA Error Status register and Fault Status 1 register for errors.
10. Check data in central memory for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Section 25 – DMA Transfer Error Test (Adapter Forced)

This section is the DMA transfer error test. Errors are forced in this section by utilizing the IPI channel adapter's hardware functions.

Subsection 0

Tests response on central memory response code parity error.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Double CMI slot and the DMA Control register force error code for invert central memory response code parity (force error code 05_{16}) in the IPI adapter DMA Control register.
5. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
6. Issue the Bus Control function 0009_{16} to Set Select Out.
7. Load the T register with the central memory address and byte count of eight (one central memory word).
8. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
9. Check, upon completion of the data transfer, the DMA Error Status register, Channel Status register, Fault Status 1 register, and Fault Status 2 register for errors.
10. Check data in central memory for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 1

Tests response on invalid response code.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Double CMI slot and the DMA Control register force error code for force invalid response code (force error code 04_{16}) in the IPI adapter DMA Control register.
5. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
6. Issue the Bus Control function 0009_{16} to Set Select Out.
7. Load the T register with the central memory address and byte count of eight (one central memory word).
8. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
9. Check, upon completion of the data transfer, the DMA Error Status register, Channel Status register, Fault Status 1 register, and Fault Status 2 register for errors.
10. Check data in central memory for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 2

Tests response on force byte count zero.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Double CMI slot and the DMA Control register Force error code for force byte count zero (Force error code 16_{16}) in the IPI adapter DMA Control register.

5. Enable and lock all transmitters and receivers, and enable Random Data Generator.
6. Issue the Bus Control function 0009_{16} to Set Select Out.
7. Load the T register with the central memory address and byte count of eight (one central memory word).
8. Initialize a central memory to IPI channel adapter interlocked mode data transfer in double octet mode.
9. Check, upon completion of the data transfer, the DMA Error Status register, Channel Status register, Fault Status 1 register, and Fault Status 2 register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```


Subsection 3

Test false start transfer, byte count zero.

Method of Execution:

1. Initialize the central memory buffer with the PP unique data pattern.
2. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
3. Execute the IPI Master Reset sequence on Port A.
4. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
5. Enable the Enable Double CMI slot in the adapter DMA Control register.
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue the Bus Control function 0009_{16} to Set Select Out.
8. Load the T register with the central memory address and byte count of zero.
9. Initiate a IPI channel to central memory adapter interlocked mode data transfer in double octet mode.
10. Check, upon completion of the data transfer, the DMA Error Status register, Channel Status register, Fault Status 1 register, and Fault Status 2 register for errors.
11. Check data in central memory for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 4

Tests force out of order.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of 16.
4. Enable the Enable Double CMI slot and the DMA Control register force error code for force out of order code (force error code 19_{16}) in the IPI adapter DMA Control register.
5. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
6. Enable the Bus Control function 0009_{16} to Set Select Out.
7. Load the T register with the central memory address and byte count of 32 (four central memory words).
8. Initiate a central memory to IPI channel adapter interlocked mode data transfer in double octet mode.
9. Check, upon completion of the data transfer, the DMA Error Status register, Channel Status register, Fault Status 1 register, and Fault Status 2 register for errors.
10. Check data in central memory for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 5

Tests force data overflow.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset Sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Double CMI slot and the DMA Control register Force error code for force data overflow code (Force error code $1B_{16}$) in the IPI adapter DMA Control register.
5. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
6. Issue the Bus Control function 0009_{16} to Set Select Out.
7. Load the T register with the central memory address and byte count of eight (one central memory word).
8. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
9. Check, upon completion of the data transfer, the DMA Error Status register, Channel Status register, Fault Status 1 register, and Fault Status 2 register for errors.
10. This subsection is repeated for data transfers in both directions.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Section 26 - DMA Transfer Error Test (MR Forced)

This section is the DMA transfer error test. Errors are forced in this section by utilizing the IOU Test Mode Maintenance register hardware functions. The section is bypassed if PARAM15 is set.

NOTE

Subsections 0, 3, 4, 5, and 6 are not executed when in concurrent maintenance mode.

Subsection 0

Test response on uncorrected write error.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset Sequence on Port A.
3. Initialize the Random Data Generator with a data pattern and burst count of four.
4. Enable the Enable Double CMI slot in the adapter DMA Control register.
5. Enable at the CMI the maintenance channel's force data out parity bit low (IOU Test Mode Maintenance register test code 37B).
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue the Bus Control function 0009_{16} to Set Select Out.
8. Load the T register with the central memory address and byte count of eight (one central memory word).
9. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
10. Check, upon completion of the data transfer, the DMA Error Status register, Fault Status 1 register, and Fault Status 2 register for errors.

This subsection consists of 12 conditions, each condition being a different pattern.

```
( optional error data )
MR TEST MODE =mm
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 1

Tests the response on uncorrected read errors.

Method of Execution:

1. Initializee the central memory buffer with the PP unique data pattern.
2. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
3. Execute the IPI Master Reset sequence on Port A.
4. Initialize the Random Data Generator with a data pattern and burst count of four.
5. Enable the Enable Double CMI slot in the adapter DMA Control register.
6. Enable at the CMI the maintenance channel's force data in parity bit low (IOU Test Mode Maintenance register test code 36B).
7. Enable and lock all transmitters and receivers, and the Random Data Generator.
8. Issue the Bus Control function 0009_{16} to Set Select Out.
9. Load the T register with the central memory address and byte count of eight (one central memory word).
10. Initiate a central memory to IPI channel adapter interlocked mode data transfer in double octet mode.
11. Check, upon completion of the data transfer, the DMA Error Status register, Channel Status register, Fault Status 1 register, and Fault Status 2 register for errors.
12. Disable force data in parity bit low at the CMI.

This subsection consists of eight conditions, each condition being a different pattern.

```
( optional error data )
MR TEST MODE =mm
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 2

Tests response on CMI response code parity error.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a data pattern and burst count of four.
4. Enable the Enable Double CMI slot in the adapter DMA Control register.
5. Enable at the CMI the maintenance channel's force response code parity error (IOU Test Mode Maintenance register test code 33B).
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue the Bus Control function 0009_{16} to Set Select Out.
8. Load the T register with the central memory address and byte count of eight (one central memory word).
9. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
10. Check, upon completion of the data transfer, the DMA Error Status register, Channel Status register, Fault Status 1 register, and Fault Status 2 register for errors.
11. Disable force response code parity error at the CMI.

```
( optional error data )
MR TEST MODE =mm
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 3

Tests response on CMI reject.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Double CMI slot in the adapter DMA Control register.
5. Enable at the CMI the maintenance channel's invert function out parity bit (IOU Test Mode Maintenance register test code 34B).
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue the Bus Control function 0009_{16} to Set Select Out.
8. Load the T register with the central memory address and byte count of eight (one central memory word).
9. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
10. Check, upon completion of the data transfer, the DMA Error Status register, Channel Status register, Fault Status 1 register, and Fault Status 2 register for errors.
11. Disable invert function out parity bit at the CMI.

```
( optional error data )
MR TEST MODE =mm
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 4

Tests response on address out parity error.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Double CMI slot in the adapter DMA Control register.
5. Enable at the CMI the maintenance channel's force address out parity bit low (IOU Test Mode Maintenance register test code 35B).
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue the Bus Control function 0009_{16} to Set Select Out.
8. Load the T register with the central memory address and byte count of eight (one central memory word).
9. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
10. Check, upon completion of the data transfer, the DMA Error Status register, Channel Status Register, Fault Status 1 register, and Fault Status 2 register for errors.
11. Disable the force address out parity bit low at the CMI.

This subsection is repeated for an address bit in each parity byte.

```
( optional error data )
MR TEST MODE =mm
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 5

Tests response on tag in/out errors.

Method of Execution:

1. Initialize the central memory buffer with the PP unique data pattern.
2. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
3. Execute the IPI Master Reset Sequence on Port A.
4. Initialize the Random Data Generator with a data pattern and burst count of four.
5. Enable the Enable Double CMI slot in the adapter DMA Control register.
6. Enable the maintenance channel's force tag in/out parity error (IOU Test Mode Maintenance register test code 32B) at the CMI.
7. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
8. Issue the Bus Control function 0009_{16} to Set Select Out.
9. Load the T register with the central memory address and byte count of eight (one central memory word).
10. Initiate a IPI channel adapter to central memory or central memory to IPI channel adapter interlocked mode data transfer in double octet mode.
11. Check, upon completion of the data transfer, the DMA Error Status register, Channel Status register, Fault Status 1 register, and Fault Status 2 register for errors.
12. Disable at the CMI force tag in/out parity error.

This subsection is repeated for transfers in both directions.

```
( optional error data )
MR TEST MODE =mm
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 6

Tests response on Mark Parity bit errors.

Method of Execution:

1. Initialize the central memory buffer with the PP unique data pattern.
2. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
3. Execute the IPI Master Reset Sequence on port A.
4. Initialize the Random Data Generator with a data pattern and burst count of four.
5. Enable the Enable Double CMI slot in the adapter DMA Control register.
6. Enable the maintenance channel's Force Mark Parity bit to zero (IOU Test Mode Maintenance register test code 31B) at the CMI.
7. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
8. Issue the Bus Control function 0009_{16} to Set Select Out.
9. Load the T register with the central memory address and byte count of eight (one central memory word).
10. Initiate a IPI channel adapter to central memory or central memory to IPI channel adapter interlocked mode data transfer in double octet mode.
11. Check, upon completion of the data transfer, the DMA Error Status register, Channel Status register, Fault Status 1 register, and Fault Status 2 register for errors.
12. Disable at CMI the Force Mark Parity bit to zero.

This subsection is repeated for transfers in both directions.

```
( optional error data )
MR TEST MODE =mm
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Section 27 – Test The Data Transfer Auto Bus Control Bit

Tests the data transfer function Auto Bus Control bit.

Subsection 0

Tests Auto Bus Control on a Write Data Transfer function.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset Sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
6. Issue a Data Transfer function ($XX8Y_{16}$) with the bit set for the Auto Bus Control to the IPI channel adapter.
7. Issue the Bus Control function 0009_{16} to Set Select Out.
8. Load the T register with the central memory address and byte count of eight (one central memory word).
9. Initiate a central memory to IPI channel adapter interlocked mode data transfer in double octet mode.
10. Check, upon completion of the data transfer, the DMA Operational Status register, IPI Error register and the DMA Error Status register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 1

Tests Auto Bus Control with data on a Write Data Transfer function.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset Sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
6. Issue a Data Transfer function ($XX8Y_{16}$) with the bits set for the Auto Bus Control and data to the IPI channel adapter.
7. Issue the Bus Control function 0009_{16} to Set Select Out.
8. Load the T register with the central memory address and byte count of eight (one central memory word).
9. Initiate a central memory to IPI channel adapter interlocked mode data transfer in double octet mode.
10. Check, upon completion of the data transfer, the DMA Operational Status register, IPI Error register and the DMA Error Status register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 2

Tests Auto Bus Control on a Read Data Transfer function.

Method of Execution:

1. Initialize the central memory address that will be read.
2. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
3. Execute the IPI Master Reset sequence on Port A.
4. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
5. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue a Data Transfer function ($XX8Y_{16}$) with the bit set for auto bus control to the IPI channel adapter.
8. Issue the Bus Control function 0009_{16} to Set Select Out.
9. Load the T register with the central memory address and byte count of eight (one central memory word).
10. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
11. Check, upon completion of the data transfer, the DMA Operational Status register, IPI Error register and the DMA Error Status register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 3

Tests Auto Bus Control with data on a Read Data Transfer function.

Method of Execution:

1. Initialize the central memory address that will be read.
2. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
3. Execute the IPI Master Reset sequence on Port A.
4. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
5. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue a Data Transfer function ($XX8Y_{16}$) with the bits set for the auto bus control and data to the IPI channel adapter.
8. Issue the Bus Control function 0009_{16} to Set Select Out.
9. Load the T register with the central memory address and byte count of eight (one central memory word).
10. Initiate a IPI channel adapter to central memory interlocked mode data transfer is in double octet mode.
11. Check, upon completion of the data transfer, the DMA Operational Status register, IPI Error register and the DMA Error Status register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Section 28 – Test The Data Transfer Auto Ending Status Bit

Tests the data transfer Auto Ending Status Bit.

Subsection 0

Tests Auto Ending Status on a Write Data Transfer function.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
6. Issue a Data Transfer function ($XX8Y_{16}$) with the bit set for the Auto Ending Status to the IPI channel adapter.
7. Issue the Bus Control function 0009_{16} to Set Select Out.
8. Load the T register with the central memory address and byte count of eight (one central memory word).
9. Initiate a central memory to IPI channel adapter interlocked mode data transfer in double octet mode.
10. Check, upon completion of the data transfer, the DMA Operational Status register, IPI Error register and the DMA Error Status register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 1

Tests Auto Ending Status on a Read Data Transfer function.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
6. Issue a Data Transfer function ($XX8Y_{16}$) with the bit set for the Auto Ending Status to the IPI channel adapter.
7. Issue the Bus Control function 0009_{16} to Set Select Out.
8. Load the T register with the central memory address and byte count of eight (one central memory word).
9. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
10. Check, upon completion of the data transfer, the DMA Operational Status register, IPI Error register and the DMA Error Status register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 2

Tests Auto Ending Status with odd octet on a Write Data Transfer function.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable and lock all transmitters and receivers, and enable the Random Data Generator.

6. Issue a Data Transfer Function ($XX8Y_{16}$) with the bits set for the Auto Ending Status and odd octet to the IPI channel adapter.

7. Issue the Bus Control function 0009_{16} to Set Select Out.

8. Load the T register with the central memory address and byte count of eight (one central memory word).

9. Initiate a central memory to IPI channel adapter interlocked mode data transfer in double octet mode.

10. Check, upon completion of the data transfer, the DMA Operational Status register, IPI Error register and the DMA Error Status register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```


Subsection 3

Tests Auto Ending Status with odd octet on a Read Data Transfer function.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.

2. Execute the IPI Master Reset sequence on Port A.

3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.

4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.

5. Enable and lock all transmitters and receivers, and enable the Random Data Generator.

6. Issue a Data Transfer function ($XX8Y_{16}$) with the bits set for the Auto Ending Status and odd octet to the IPI channel adapter.

7. Issue the Bus Control function 0009_{16} to Set Select Out.

8. Load the T register with the central memory address and byte count of eight (one central memory word).

9. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.

10. Check, upon completion of the data transfer, the DMA Operational Status register, IPI Error register and the DMA Error Status register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 4

Tests Auto Ending Status with Status Terminate on a Write Data Transfer function.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
6. Issue a Data Transfer function ($XX8Y_{16}$) with the bits set for the Auto Ending Status and Status Terminate to the IPI channel adapter.
7. Issue the Bus Control function 0009_{16} to Set Select Out.
8. Load the T register with the central memory address and byte count of eight (one central memory word).
9. Initiate a central memory to IPI channel adapter interlocked mode data transfer in double octet mode.
10. Check, upon completion of the data transfer, the DMA Operational Status register, IPI Error register and the DMA Error Status register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 5

Tests Auto Ending Status with Status Terminate on a Read Data Transfer function.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
6. Issue a Data Transfer function ($XX8Y_{16}$) with the bits set for the Auto Ending Status and Status Terminate to the IPI channel adapter.
7. Issue the Bus Control function 0009_{16} to Set Select Out.
8. Load the T register with the central memory address and byte count of eight (one central memory word).
9. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
10. Check, upon completion of the data transfer, the DMA Operational Status register, IPI Error register and the DMA Error Status register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Section 29 – Pause Test

This section is the Pause test.

Subsection 0

Tests pause on Force Short Pause.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable Force Short Pause in the IPI channel adapter using the Force Error function (3022_{16}).
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue a Data Transfer function ($XX8Y_{16}$) with the bits set for Auto Master/Slave status and Status Terminate bits to the IPI channel adapter.
8. Issue the Bus Control function 0009_{16} to Set Select Out.
9. Load the T register with the central memory address and byte count of eight (one central memory word).
10. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
11. Check, upon completion of the data transfer, the Bus B and status of the channel is read using the IPI Bus Control function 0041_{16}.
12. Read and check the DMA Operational Status register, and the DMA Error Status register and Fault Status 1 register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 1

Tests pause on Force Short Pause and Odd Octet Transfer.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable Force Short Pause in the IPI channel adapter using the Force Error function (3022_{16}).
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue a Data Transfer function ($XX8Y_{16}$) with the bits set for Auto Master/Slave status, Odd Octet Transfer and Status Terminate bits to the IPI channel adapter.
8. Issue the Bus Control function 0009_{16} to Set Select Out.
9. Load the T register with the central memory address and byte count of eight (one central memory word).
10. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
11. Check, upon completion of the data transfer, the Bus B and status of the channel is read using the IPI Bus Control function 0041_{16}.
12. Read and check the DMA Operational Status register, and the DMA Error Status register and Fault status 1 register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 2

Tests the Pause flag on Force Short Pause.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset Sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Pause Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable Force Short Pause in the IPI channel adapter using the Force Error function (3022_{16}).
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue a Data Transfer function ($XX8Y_{16}$) with the bits set for Auto Master/Slave Status and Status Terminate bits to the IPI channel adapter.
8. Issue the Bus Control function 0009_{16} to Set Select Out.
9. Load the T register with the central memory address and byte count of eight (one central memory word).
10. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
11. Check, upon completion of the data transfer, the Bus B and status of the channel is read using the IPI Bus Control function 0041_{16}.
12. Read and check the DMA Operational Status register, the channel flag for being set, and the DMA Error Status register and Fault status 1 register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 3

Tests Pause on Force Long Pause.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable Force Long Pause in the IPI channel adapter using the Force Error function (4022_{16}).
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue a Data Transfer function ($XX8Y_{16}$) with the bits set for Auto Master/Slave Status and Status Terminate bits to the IPI channel adapter.
8. Issue the Bus Control function 0009_{16} to Set Select Out.
9. Load the T register with the central memory address and byte count of eight (one central memory word).
10. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
11. Check, upon completion of the data transfer, the Bus B and status of the channel is read using the IPI Bus Control function 0041_{16}.
12. Read and check the DMA Operational Status register, and the DMA Error Status register and Fault Status 1 register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 4

Tests Pause on Force Long Pause and Odd Octet Transfer.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable Force Long Pause in the IPI channel adapter using the Force Error function (4022_{16}).
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue a Data Transfer function ($XX8Y_{16}$) with the bits set for Auto Master/Slave Status, Odd Octet Transfer, and Status Terminate bits to the IPI channel adapter.
8. Issue the Bus Control function 0009_{16} to Set Select Out.
9. Load the T register with the central memory address and byte count of eight (one central memory word).
10. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
11. Check, upon completion of the data transfer, the Bus B and status of the channel is read using the IPI Bus Control function 0041_{16}.
12. Read and check the DMA Operational Status register, and the DMA Error Status register and Fault Status 1 register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 5

Tests resume data transfer after pause.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Pause Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable Force Short Pause in the IPI channel adapter using the Force Error function (3022_{16}).
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue a Data Transfer function ($XX8Y_{16}$) with the bits set for Auto Master/Slave Status and Status Terminate bits to the IPI channel adapter.
8. Issue the Bus Control function 0009_{16} to Set Select Out.
9. Load the T register with the central memory address and byte count of eight (one central memory word).
10. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
11. Check, upon completion of the first data transfer.
12. Initiate a second IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
13. Check, upon completion of the first data transfer, the Bus B and status of the channel is read using the IPI Bus Control function 0041_{16} and then checked, and the DMA Operational Status register is read and checked.
14. Issue a Data Transfer function ($XX8Y_{16}$) with the bits set for Auto Master/Slave Status and Status Terminate bits to the IPI channel adapter.
15. Check, upon completion of the second data transfer, the Bus B and status of the channel is read using the IPI Bus Control function 0041_{16}.
16. Read and check the DMA Operational Status register, and the DMA Error Status register and Fault Status 1 register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Section 30 – Test The 1022H and 2022H Force Error Functions

Tests the Force Parity Error On Status function (1022_{16}) and Force Terminate Status function (2022_{16}).

Subsection 0

Tests Force Parity Error On Status.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable Force Parity Error On Status in the IPI channel adapter using the Force Error function (1022_{16}).
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue a Data Transfer function ($XX8Y_{16}$) with the bits set for Auto Master/Slave Status and Status Terminate bits to the IPI channel adapter.
8. Issue the Bus Control function 0009_{16} to Set Select Out.
9. Load the T register with the central memory address and byte count of eight (one central memory word).
10. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
11. Check, upon completion of the data transfer, the Bus B and status of the channel is read using the IPI Bus Control function 0041_{16}.
12. Read and check the DMA operational status register, and the DMA Error Status register and Fault Status 1 register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 1

Tests Force Terminate Status.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable Force Terminate Status in the IPI channel adapter using the Force Error function (2022_{16}).
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue a Data Transfer function ($XX8Y_{16}$) with the bits set for Auto Master/Slave Status and Status Terminate bits to the IPI channel adapter.
8. Issue the Bus Control function 0009_{16} to Set Select Out.
9. Load the T register with the central memory address and byte count of eight (one central memory word).
10. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
11. Check, upon completion of the data transfer, the Bus B and status of the channel is read using the IPI Bus Control function 0041_{16}.
12. Read and check the DMA Operational Status register, and the DMA Error Status register and Fault Status 1 register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 2

Tests Force Terminate Status and Odd Octet Transfer.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable Force Terminate Status in the IPI channel adapter using the Force Error function (2022_{16}).
6. Enable and lock all transmitters and receivers and enable the Random Data Generator.
7. Issue a Data Transfer function ($XX8Y_{16}$) with the bits set for Auto Master/Slave Status, Odd Octet Transfer and Status Terminate bits to the IPI channel adapter.
8. Issue the Bus Control function 0009_{16} to Set Select Out.
9. Load the T register with the central memory address and byte count of eight (one central memory word).
10. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
11. Check, upon completion of the data transfer, the Bus B and status of the channel is read using the IPI Bus Control function 0041_{16}.
12. Read and check the DMA Operational Status register, and the DMA Error Status register and Fault Status 1 register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 3

Tests Force Terminate Status without a Status Terminate.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable Force Terminate Status in the IPI channel adapter using the Force Error function (2022_{16}).
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue a Data Transfer function ($XX8Y_{16}$) with the bit set for Auto Master/Slave Status to the IPI channel adapter.
8. Issue the Bus Control function 0009_{16} to Set Select Out.
9. Load the T register with the central memory address and byte count of eight (one central memory word).
10. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
11. Check, upon completion of the data transfer, the Bus B and status of the channel is read using the IPI Bus Control function 0041_{16} and then checked.
12. Read and check the DMA Operational Status register, and the DMA Error Status register and Fault Status 1 register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Section 31 – Test The 5022H, 6022H, and 7022H Force Error Functions

Tests the Force Pause Terminate function (5022_{16}), Force Unsuccessful Transfer function (6022_{16}), and Force Parity function (7022_{16}).

Subsection 0

Tests Force Pause Terminate.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable Force Pause Terminate in the IPI channel adapter using the Force Error function (5022_{16}).
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue a Data Transfer function ($XX8Y_{16}$) with the bits set for Auto Master/Slave Status and Status Terminate bits to the IPI channel adapter.
8. Issue the Bus Control function 0009_{16} to Set Select Out.
9. Load the T register with the central memory address and byte count of eight (one central memory word).
10. Initiate a central memory to IPI channel adapter interlocked mode data transfer in double octet mode.
11. Check, upon completion of the data transfer, the Bus B and status of the channel is read using the IPI Bus Control function 0041_{16} and then checked.
12. Read and check the DMA Operational Status register, and the DMA Error Status register and Fault Status 1 register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 1

Tests Force Pause Terminate and Odd Octet Transfer.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable Force Pause Terminate in the IPI channel adapter using the Force Error function (5022_{16}).
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue a Data Transfer function ($XX8Y_{16}$) with the bits set for Auto Master/Slave Status, Odd Octet Transfer and Status Terminate bits to the IPI channel adapter.
8. Issue the Bus Control function 0009_{16} to Set Select Out.
9. Load the T register with the central memory address and byte count of eight (one central memory word).
10. Initiate a central memory to IPI channel adapter interlocked mode data transfer in double octet mode.
11. Check, upon completion of the data transfer, the Bus B and status of the channel is read using the IPI Bus Control function 0041_{16}.
12. Read and check the DMA Operational Status register, and the DMA Error Status register and Fault Status 1 register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 2

Tests Force Unseccessful Transfer.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence is executed on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable Force Unsuccessful Transfer in the IPI channel adapter using the Force Error function (6022_{16}).
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue a Data Transfer function ($XX8Y_{16}$) with the bits set for Auto Master/Slave Status and Status Terminate bits to the IPI channel adapter.
8. Issue the Bus Control function 0009_{16} to Set Select Out.
9. Load the T register with the central memory address and byte count of eight (one central memory word).
10. Initiate a central memory to IPI channel adapter interlocked mode data transfer in double octet mode.
11. Check, upon completion of the data transfer, the Bus B and status of the channel is read using the IPI Bus Control function 0041_{16}.
12. Read and check the DMA Operational Status register, and the DMA Error Status register and Fault Status 1 register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 3

Tests Force Unsuccessful Transfer and Odd Octet Transfer.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence is executed on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable Force Unsuccessful Transfer in the IPI channel adapter using the Force Error function (6022_{16}).
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue a Data Transfer function ($XX8Y_{16}$) with the bits set for Auto Master/Slave Status, Odd Octet Transfer and Status Terminate bits to the IPI channel adapter.
8. Issue the Bus Control function 0009_{16} to Set Select Out.
9. Load the T register with the central memory address and byte count of eight (one central memory word).
10. Initiate a central memory to IPI channel adapter interlocked mode data transfer in double octet mode.
11. Check, upon completion of the data transfer, the Bus B and status of the channel is read using the IPI Bus Control function 0041_{16}.
12. Read and check the DMA Operational Status register, and the DMA Error Status register and Fault Status 1 register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 4

Tests Force Parity.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable Force Parity in the IPI channel adapter using the Force Error function (7022_{16}).
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue a Data Transfer function ($XX8Y_{16}$) with the bits set for Auto Master/Slave Status and Status Terminate bits to the IPI channel adapter.
8. Issue the Bus Control function 0009_{16} to Set Select Out.
9. Load the T register with the central memory address and byte count of eight (one central memory word).
10. Initiate a central memory to IPI channel adapter interlocked mode data transfer in double octet mode.
11. Check, upon completion of the data transfer, the Bus B and status of the channel is read using the IPI Bus Control function 0041_{16}.
12. Read and check the DMA Operational Status register, and the DMA Error Status register and Fault Status 1 register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 5

Tests Force Parity and Odd Octet Transfer.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of four.
4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable Force Parity in the IPI channel adapter using the Force Error function (7022_{16}).
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue a Data Transfer function ($XX8Y_{16}$) with the bits set for Auto Master/Slave Status, Odd Octet Transfer and Status Terminate bits to the IPI channel adapter.
8. Issue the Bus Control function 0009_{16} to Set Select Out.
9. Load the T register with the central memory address and byte count of eight (one central memory word).
10. Initiate a central memory to IPI channel adapter interlocked mode data transfer in double octet mode.
11. Check, upon completion of the data transfer, the Bus B and status of the channel is read using the IPI Bus Control function 0041_{16}.
12. Read and check the DMA Operational Status register, and the DMA Error Status register and Fault Status 1 register for errors.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Section 32 - DMA Transfer Test

This section is the DMA transfer test.

Subsection 0

Tests incomplete word DMA read transfers in interlocked mode.

Method of Execution:

1. Initialize the central memory address that will be read.
2. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
3. Execute the IPI Master Reset sequence on Port A.
4. Initialize the Random Data Generator with a PP unique data pattern and burst count of one.
5. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue the Bus Control function 0009_{16} to Set Select Out.
8. Load the T register with the central memory address and byte count of one.
9. Initiate a central memory to IPI channel adapter interlocked mode data transfer in double octet mode.
10. Check, upon completion of the data transfer, the DMA Error Status register and Fault Status 1 register for errors.

This subsection consists of 16 conditions, each a different byte count from 1 to 16.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 1

Tests incomplete word DMA write transfers in interlocked mode.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a PP unique data pattern and burst count of one.
4. Enable the Enable Tip Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
6. Issue the Bus Control function 0009_{16} to Set Select Out.
7. Load the T-register with the central memory address and byte count of one.
8. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
9. Check, upon completion of the data transfer, the DMA Error Status register and Fault Status 1 register for errors.
10. Check data in central memory for errors.

This subsection consists of 16 conditions, each a different byte count from 1 to 16.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 2

Tests DMA transfers in interlocked mode.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a random data pattern and burst count for the maximum number of central memory words available times four.
4. Enable the Enable T Prime Empty Flag and Enable Double CMI slot in the adapter DMA Control register.
5. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
6. Issue the Bus Control function 0009_{16} to Set Select Out.
7. Load the T register with the central memory address and byte count for the maximum number of central memory words available.
8. Initiate a IPI channel adapter to central memory interlocked mode data transfer in double octet mode.
9. Check, upon completion of the data transfer, the DMA Error Status register and Fault Status 1 register for errors.
10. Master Clear the IPI channel adapter.
11. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
12. Execute the IPI Master Reset sequence on Port A.
13. Initialize the Random Data Generator with a random data pattern and burst count for the maximum number of central memory words available times four.
14. Enable the Enable T Prime Empty Flag and Enable Double CMI slot in the adapter DMA Control register.
15. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
16. Issue the Bus Control function 0009_{16} to Set Select Out.

17. Load the T register with the central memory address and byte count for the maximum number of central memory words available.

18. Initiate a central memory to IPI channel adapter interlocked mode data transfer in double octet mode.

19. Check, upon completion of the data transfer, the DMA Error Status register and Fault Status 1 register for errors.

20. Read and check the vector in the Test Mode Operand Generator.

This subsection consists of 10 conditions, each being a different random data pattern.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Subsection 3

Tests DMA transfers in streaming mode.

Method of Execution:

Same as Subsection 2, but transfers are in streaming mode.

```
( optional error data )
TEST SEED ssss
XFER blk bbbb tttt tttt
```

Section 33 – Stacked Transfer Register Test

This section is the stacked transfer test.

NOTE

This section is not executed when in concurrent maintenance mode.

Subsection 0

A minimum of six transfers will be sequentially sent to the IPI channel adapter. The byte counts are generated randomly. The starting real memory addresses are computed from the byte counts.

Method of Execution:

1. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
2. Execute the IPI Master Reset sequence on Port A.
3. Initialize the Random Data Generator with a random data pattern and burst count equal to the transfer byte count divided by two for the first data transfer.
4. Enable the Enable T Prime Empty Flag and Enable Double CMI slot in the adapter DMA control register.
5. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
6. Load the T register with the central memory address and byte count for the current transfer.
7. Issue the Bus Control function 0009_{16} to Set Select Out.
8. Initiate adapter to central memory interlocked and streaming mode data transfers in double octet mode until the byte count/address pairs have expired.
9. Master Clear the IPI adapter channel.
10. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
11. Execute the IPI Master Reset Sequence on Port A.
12. Initialize the Random Data Generator with a random data pattern and burst count equal to the transfer byte count divided by two for the first data transfer.
13. Enable the Enable T Prime Empty Flag and Enable Double CMI slot in the adapter DMA Control register.
14. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
15. Load the T register with the central memory address and byte count for the current transfer.
16. Issue the Bus Control function 0009_{16} to Set Select Out.

17. Initiate central memory to IPI channel adapter interlocked and streaming mode data transfers in double octet mode with the same byte count/address pairs to check the data in CM.

18. Check, upon completion of the data transfer, DMA Error Status registers and the Fault Status 1 for errors.

This sequence is repeated for all combinations of interlocked and streaming transfers.

```
( optional error data )
TEST SEED ssss
XFER no. nn
nxt cccc bbbb tttt tttt
cur cccc bbbb tttt tttt
```

Section 34 - DMA Transfer Timing Test

This section is the DMA transfer test. The user must enable the section by setting the bit for the section in parameter word 7.

NOTE

This section is not executed when in concurrent maintenance mode.

Subsection 0

This subsection checks the data transfer rate between the IPI channel adapter and central memory.

Method of execution:

1. Initialize the central memory buffer with a random data pattern.
2. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
3. Execute the IPI Master Reset sequence on Port A.
4. Initialize the Random Data Generator with a random data pattern and burst count of 8000_{16}.
5. Enable the Enable T Prime Empty Flag and Enable Double CMI slot in the adapter DMA control register.
6. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
7. Issue the Bus Control function 0009_{16} to Set Sync Out.
8. Load the T register with the central memory address and byte count of $FFFF_{16}$.
9. Initiate a central memory-to-adapter streaming mode data transfer in double octet mode.
10. Determine the amount of time required to do the DMA data transfer.
11. Master Clear the IPI adapter channel.
12. Set the sync period to the 25-MHz internal clock to select the 25-megabyte transfer rate.
13. Execute the IPI Master Reset sequence on Port A.
14. Initialize the Random Data Generator with a random data pattern and burst count of 8000_{16}.
15. Enable the Enable T Prime Empty Flag in the adapter DMA control register.
16. Enable and lock all transmitters and receivers, and enable the Random Data Generator.
17. Issue the Bus Control function 0009_{16} to Set Sync Out.

18. Load the T register with the central memory address and byte count of $FFFF_{16}$.

19. Initiate a central memory-to-adapter streaming mode data transfer in double octet mode.

20. Determine the amount of time required to perform the DMA data transfer.

21. If the DMA data transfer rate with the Double CMI slot enabled was greater than or equal to 20 MHz, display the message `DMA TRANSFER TIMING PASSED`. If the DMA data transfer rate was less than 20 MHz, display the message `DMA TRANSFER TIMING FAILED`.

Glossary A

Glossary A

A

ABS
Absolute.

ACU
Address Control.

AC1
Address Control Unit 1.

AC2
Address Control Unit 2.

ADJ
Adjuster.

ADRS
Address.

ALU
Arithemtic and Logical Unit.

AOR
Address Out of Range.

ARITH
Arithmetic.

ASCII
American Standard Code for Information Interchange.

ASE
Address Specificaton Error.

ASID
Active Segment Identifier.

ASSY
Assembly.

ASYNC
Asynchronous.

AUX
Auxiliary.

AVAL
Available.

B

BC
Base-Constant.

BCD
Binary Coded Decimal.

BD
Branch Delta.

BDP
Business Data Processor.

BFR
Buffer.

BIN
Binary.

BKPT
Breakpoint.

BN
Byte Number.

BOL
Boolean Unit.

Bound
Boundary.

BR
Branch.

BSP
Binding Section Pointer.

C

CA
Condition Action.

CABR
Cache Address Buffer Register.

CAE
Condition Action Extension.

CAR
Cache Address Register.

CBP
Code Base Pointer.

CCN
Conflict Control Network.

CCR
Clock Condition Register.

CEL
Corrected Error Log.

CFF
Critical Frame Flag.

CHAN
Channel.

CIR
Current Instruction Register.

CLK
Clock.

CLR
Clear.

CM
Central Memory.

CMC
Central Memory Control.

CMI
Central Memory Interface

CMP
Compare Unit.

CMPR
Compare.

CM1
Central Memory Control Unit 1.

CM2
Central Memory Control Unit 2.

CM3
Central Memory Control Unit 3.

CM4
Central Memory Control Unit 4.

CNT
Count.

CNTR
Counter.

COEF
Coefficient.

COM
Common.

COMP
Complement.

COMPL
Complete.

COND
Condition.

CONFL
Conflict.

CONT
Control.

CONV
Convert, Converter.

COR
Corrected.

CP
Clock Period.

CPU
Central Processing Unit.

CS
Condition Sensing.

CS/CW
Control Store/Control Word.

CSA
Control Store Address.

CSD
Control Store Data.

CST
Control Store.

CSU
Control Storage Unit.

CW
Control Word.

CWA
Control Word Address.

CWD
Control Word Data.

D

DCDR
Decoder.

DEC
Dependent Environment Control, Decimal.

DECR
Decrement.

DESCR
Descriptor.

DESTN
Destination.

DET
Detect, Detected.

DI
Debug Index.

DISASSY
Disassembly.

DISTR
Distributor.

DIV
Divide.

DLP
Debug List Pointer.

DLY
Delay.

DLYD
Delayed.

DM
Debug Mask.

DMR
Debug Mask Register.

DSBL
Disable.

DUE
Detected Uncorrected Error.

E

EBCDIC
Extended Binary Coded Decimal Interchange Code.

ECC
Error Correction Code.

EI
Environment Interface.

EIT
Error Information Table.

EMMR
Exit Mode Mask Register.

ENBL
Enable.

ENCDR
Encoder.

ENV
Environment.

EPN
Error Processing Network.

ESE
Environment Specification Error.

EXCH
Exchange.

EXP
Exponent.

EXT
External, Extension.

F

FCTN
Function.

FF
Flip Flop.

FIFO
First In, First Out.

FLC
C170 Mode Central Memory Field Length.

FNO
Fanout.

FP
Floating Point.

FPU
Floating Point Unit.

FRC
Free Running Counter.

FSM
Finite State Machine.

FU
Functional Unit.

G

GEN
Generator.

GENL
General.

GRP
Group.

GT
Greater Than.

GTE
Greater Than or Equal To.

H

HF
History File.

HLDG
Holding.

HT
History Tag.

IAD
Integer Add Unit.

IBS
Instruction Buffer Stack.

ID
Indentification.

IDU
Instruction Decode Unit.

IDX
Indexed.

IGU
Integer Unit.

II
Instruction Issue.

IMMED
Immediate.

IMU
Integer Multiply Unit.

INC
Increment Unit.

INCR
Increment.

INDEF
Indefinite.

INF
Infinite.

INH
Inhibit.

INIT
Initial.

INP
Input.

INSTR
Instruction.

INT
Internal.

INTERSEG
Intersegment.

INTRPT
Interrupt.

INVAL
Invalid.

IN1
Instruction Unit 1.

IN2
Instruction Unit 2.

IOU
Input/Output Unit.

IPR
Instruction Parcel Register.

ISE
Instruction Specification Error.

J

JPS
Job Process State Pointer.

K

KC
Keypoint Code.

KCN
Keypoint Class Number.

KM
Keypoint Mask.

KYPT
Keypoint.

L

LENG
Length.

LOS
Loss of Significance.

LPTR
Last Packet to Return.

LRU
Least Recently Used.

LSD
Least Significant Digit.

LSI
Large Scale Inegration.

LSM
Last Symbol Mask.

LSU
Load/Store Unit.

LT
Less Than.

LTE
Less Than or Equal To.

LWR
Lower.

M

MAC
Maintenance Access Control.

MAINT
Maintenance.

MAR
Micrand Address Register.

MAX
Maximum.

MB

Move Bytes.

MBE

Multiple Bit Error.

MC

Master Clear.

MCH

Maintenance Channel.

MCR

Monitor Condition Register.

MDF

Model Dependent Flag.

MDW

Model Dependent Word.

MEM

Memory.

MIC

Common subroutine used by test to initiate, execute, and monitor execution of a micrand sequence.

MICR

Micrand.

MISC

Miscellaneous.

MM

Monitor Mask.

MMR

Monitor Mask Register.

MOP

Micro-Operation.

MPS

Monitor Process State Pointer.

MSB

Most Significant Bit.

MSC

Micrand Sequence Control.

MSD
Most Significant Digit.

MSNZB
Most Significant Non-Zero Byte.

MULT
Multiply.

MUX
Multiplexer.

M1
Micrand 1.

N

NEG
Negative.

NET
Network.

NO
Number.

NO-OP
No-op, No operation.

NORM
Normalize.

NS
Nanosecond

O

OCA
Operand Cache.

OCF
On Condition Flag.

OP
Operand.

OPCODE
Operation Code.

OPN
Operation.

ORIG
Original.

OSC
Oscillator.

OUT
Output.

OVFL
Overflow.

P

P
Program Address.

PAR
Parity.

PDM
Processor Detected Malfunction.

PE
Parity Error.

PFA
Page Frame Address.

PFS
Processor Fault Status.

PH
Phase.

PIT
Process Interval Timer.

PM
Page Map.

PMF
Performance Monitoring Facility.

PN
Page Number.

PO
Page Offset.

PONR
Point of No Return.

POS
Positive.

PP
Peripheral Processor.

PRED
Predicted, Predictor.

PREV
Previous.

PRI
Priority.

PRIV
Privilege.

PROC
Processor, Processing.

PROD
Product.

PSM
Page Size Mask.

PSR
Processor State Register.

PSWF
Page Table Search Without Find.

PTA
Page Table Address.

PTE
Page Table Entry.

PTL
Page Table Length.

PTM
Processor Test Mode.

PVA
Process Virtual Address.

PW
Partial Write.

PWC
PP Word Count

Q

QTY
Quantity.

QUAD
Quadrant.

R

R/W
Read/Write.

RA/FL
Reference Address/Field Length.

RAC
C170 Mode Central Memory Reference Address.

RAM
Random Access Memory.

RCL
Read and Clear Lock.

RCVR
Receiver.

RDS
Register/Data Select.

REF
Reference.

REGEN
Regenerator.

REL
Relative.

REM
Remainder.

REQ
Request.

RET
Returned.

RESYNC
Resynchronize, Resynchronizer, Resynchronization.

RF
Register File.

RGTR
Register.

RGU
Register Unit.

RMA
Real Memory Address.

RNI
Read Next Instruction.

ROM
Read Only Memory.

RPL
Result Pipeline.

RSL
Read and Set Lock.

S

SBE
Single Bit Error.

SCM
Soft Control Memory.

SCT
Special Character Table.

SCU
Scalar Unit.

SDE
Segment Descriptor Table Entry.

SECDED
Single Error Correction/Double Error Detection.

SEG
Segment.

SEL
Select.

SEP
Separate.

SEQ
Sequence.

SFDA
Stack Frame Save Area.

SH8
Shift 8.

SHF
Shift.

SIG
Significant.

SIT
System Interval Timer.

SM
Segment Map.

SPEC
Specification.

SPID
Segment/Page Identifier.

SS
Status Summary.

ST
Store.

STA
Segment Table Address.

STD
Standard.

STL
Segment Table Length.

STR
Stream.

SUBTR
Subtract.

SUPPR
Suppression.

SV
Specification Value.

SVA
System Virtual Address.

SW
Switch.

SYN
Syndrome.

SYNC
Sychronize.

SYS
System.

T

TE
Trap Enable.

TED
Trap Enable Delay.

TERMN
Terminate.

TFA
Tag File Address.

TP
Trap Pointer.

U

ubit
Micrand Bit.

ucode
Microcode.

UCR
User Condition Register.

UM
User Mask.

UMR
User Mask Register.

UNBR
Unbranch.

UNCOND
Unconditional.

UNCOR
Uncorrectable.

UNDFL
Underflow.

UNLOG
Unlogged.

UPR
Upper.

usec
Microsecond.

UTP
Untranslatable Pointer.

V

VMID
Virtual Machine Identifier.

W

WDS
Write Data Select.

WOI
Word of Interest.

WR
Write.

X

XFER
Transfer.

XLTR
Translator.

XMTR
Transmitter.

XOR
Exclusive OR.

A

Z

ZF

Zero Flag.

Comments (continued from other side)

Please fold on dotted line;
seal edges with tape only.

FOLD

FOLD

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
First-Class Mail Permit No. 4760 St. Paul, MN

POSTAGE WILL BE PAID BY ADDRESSEE

CONTROL DATA
Technical Publications ARH219
4201 Lexington Avenue N.
St. Paul, MN 55126-9983

We would like your comments on this manual to help us improve it. Please take a few minutes to fill out this form.

Who are you?

- ☐ Manager
- ☐ Systems analyst or programmer
- ☐ Applications programmer
- ☐ Operator
- ☐ Other _____

How do you use this manual?

- ☐ As an overview
- ☐ To learn the product or system
- ☐ For comprehensive reference
- ☐ For quick look-up
- ☐ Other _____

What programming languages do you use? _____

_____

How do you like this manual? Answer the questions that apply.

Yes	Somewhat	No	
☐	☐	☐	Does it tell you what you need to know about the topic?
☐	☐	☐	Is the technical information accurate?
☐	☐	☐	Is it easy to understand?
☐	☐	☐	Is the order of topics logical?
☐	☐	☐	Can you easily find what you want?
☐	☐	☐	Are there enough examples?
☐	☐	☐	Are the examples helpful? (☐ Too simple? ☐ Too complex?)
☐	☐	☐	Do the illustrations help you?
☐	☐	☐	Is the manual easy to read (print size, page layout, and so on)?
☐	☐	☐	Do you use this manual frequently?

Comments? If applicable, note page and paragraph. Use other side if needed.

Check here if you want a reply: ☐

Name _____ Company _____

Address _____ Date _____

_____ Phone _____

Please send program listing and output if applicable to your comment.